

---

Universidad Carlos III de Madrid  
Campus de Colmenarejo

---

Ingeniería Técnica en Informática de Gestión



Proyecto de Fín de Carrera

# Muusic:

mashup de servicios web musicales

---

Alumno:	Félix Manuel Lamazares Montes
Tutor:	César de Pablo Sánchez

---

Noviembre de 2008



## Agradecimientos

---

Quiero agradecer el apoyo de mi familia durante todos mis estudios: a mis padres Manolo y Maria Jesús; y a mi hermana Paloma. Gracias por ayudarme con la carrera. Gracias a mi padre por leer toda la memoria para corregir errores ortográficos.

Gracias a mi tutor César que me ha ayudado mucho; y me ha dado ánimo en momentos de estrés.

Gracias también a Pilar y a todos mis amigos que siempre están ahí.

# 0. Índice

0. Índice .....	4
1. Introducción.....	6
1.a. Prefacio .....	7
1.a.i. Multimedia.....	7
1.a.ii. Servicios Web.....	7
1.b. Ámbito.....	8
1.c. Objetivos.....	9
2. Estado del arte .....	10
2.a. Integración de aplicaciones heterogéneas (mashups) .....	12
2.a.i. Tipos de mashups atendiendo al contenido .....	12
2.a.ii. Tipos de mashup atendiendo a su localización.....	13
2.a.iii. Tecnología .....	14
2.b. Servicios Web.....	18
2.b.i. RPC .....	18
2.b.ii. SOAP.....	19
2.b.iii. REST .....	20
2.c. Servicios web orientados a la música .....	22
2.c.i. Last.fm .....	22
2.c.ii. Foafing the music .....	24
2.c.iii. MyStrands .....	25
2.c.iv. Google Music Trends .....	27
2.c.v. MySpace .....	27
2.c.vi. Grooveshark.....	28
2.d. Fuentes de información .....	30
2.d.i. Buscadores de música .....	30
2.d.ii. Wikipedia .....	32
2.d.iii. Letras de canciones .....	32
2.d.iv. Medios .....	34
2.d.v. Buscadores de vídeos .....	36
2.d.vi. Agendas de conciertos.....	37
2.e. Tecnología .....	39
2.e.i. Twitter .....	39
2.e.ii. Yahoo! Pipes .....	39
2.e.iii. Songbird .....	40
2.f. Extracción de contenido estático (scraping).....	43
2.f.i. Dapper.....	43
2.f.ii. OpenKapow RoboMaker .....	44
3. Aplicación.....	45
3.a. Introducción.....	46
3.b. Arquitectura.....	47
3.c. Descargar canciones con Yahoo! Pipes .....	48

3.c.i. Entorno de desarrollo.....	48
3.c.ii. Requisitos .....	49
3.c.iii. Solución.....	50
3.d. Creando un feed con RoboMaker.....	59
3.d.i. Requisitos .....	59
3.d.ii. Entorno de desarrollo .....	60
3.d.iii. Solución.....	61
3.e. Integración con Songbird.....	62
3.e.i. Requisitos .....	62
3.e.ii. Solución.....	63
4. Discusión .....	67
4.a. Introducción.....	68
4.b. Yahoo! Pipes .....	69
4.c. Screen scraping con RoboMaker .....	71
4.d. Songbird .....	72
5. Conclusiones.....	73
5.a. Introducción.....	74
5.b. Líneas futuras de desarrollo.....	76
5.b.i. Desambiguación .....	76
5.b.ii. Conciertos por localización .....	76
5.b.iii. Detección de errores en peticiones HTTP.....	77
5.b.iv. Redifusión de temas en Twitter.....	78
6. Bibliografía.....	79
7. Anexos.....	84
7.a. Instalación de la extensión para Songbird .....	85
7.b. Atribuciones .....	87

# 1 ■ Introducción

## 1.a. Prefacio

---

La World Wide Web fue creada en 1989 por Tim Berners-Lee y Robert Cailliau. Desde entonces han pasado diecinueve años, durante los cuales “la Web” ha sufrido grandes cambios, tanto en objetivos como en contenido.

Comenzó siendo utilizada como una herramienta colaborativa, de inteligencia colectiva. Se le fueron encontrando otros usos como plataforma para noticias, bitácora, páginas personales, páginas empresariales, etc. Poco a poco fue aumentando la diversidad de contenidos que se encontraban por la Web, evolucionando desde simples documentos de hipertexto a contenido multimedia, como imágenes, audio o vídeo.

### *1.a.i. Multimedia*

Con el paso del tiempo, el contenido multimedia que se puede encontrar en la Web ha crecido de forma exponencial, de forma que existen multitud de herramientas, protocolos, servidores donde encontrar ingentes cantidades de recursos multimedia. Se puede decir que la Web se ha convertido en uno de los principales repositorios multimedia del mundo.

La evolución del contenido multimedia y de la música en la Web ha propiciado la creación de nuevos servicios que ofrecen información orientada a estos contenidos, así como posibilidades para conocer nuevos contenidos relacionados con los gustos del usuario, a través de ciertas relaciones que dichos servicios establecen entre los contenidos.

### *1.a.ii. Servicios Web*

Al tiempo que han ido surgiendo ciertos servicios, ha surgido a su vez la necesidad de interoperabilidad entre dichos servicios. Es aquí donde aparece la figura de los Servicios Web, un conjunto de herramientas destinadas a permitir el intercambio de datos entre aplicaciones distribuidas en diferentes puntos de una red.

Distintas aplicaciones creadas con distintos lenguajes de programación pueden utilizar estas herramientas para intercambiar datos entre sí a través de Internet.

## 1.b. Ámbito

---

Se tratarán dos temas principales en este proyecto: los **servicios musicales** que se encuentran disponibles a través de Internet, y la **creación de mashups** o aplicaciones heterogéneas.

Al hablar de servicios web musicales nos referimos a las aplicaciones web que de alguna manera tienen que ver con la música:

- Sistemas de recomendación musical.
- Radio por Internet.
- Buscadores de música.
- Fuentes de información relacionadas con la música, como:
  - Webs con letras de canciones.
  - Webs con información de artistas, discos, etc.
  - Agendas de conciertos.
- Medios:
  - Revistas especializadas.
  - Blogs de opinión.
  - Portales musicales.

Por otro lado, un “mashup” es un tipo de aplicación web que combina el contenido de otras dos o más aplicaciones web, para crear un nuevo contenido completo. Se hablará sobre las tecnologías existentes, su arquitectura y las distintas posibilidades que ofrecen.

Finalmente se buscará un nexo de unión entre ambos puntos, de forma que se creará una aplicación musical basándose en la información que ofrecen otras aplicaciones musicales que utilizaremos como fuentes primarias de información para la creación de nuestro mashup.

Además de estos dos grandes puntos, también se tocan otros aspectos, como la extracción de contenido estático o screen scraping. Es una técnica que extrae contenido sólo legible por humanos de una página web estática y lo convierte a contenido legible también por máquinas, algo que le confiere la cualidad de ser reutilizable por otras aplicaciones.



## 1.c. Objetivos

---

Una vez definido el ámbito de nuestro proyecto, hablaremos de los objetivos:

- Descubrir los **servicios** que se ofrecen en Internet relacionados con el **ámbito musical**, tanto directa como indirectamente. Analizar las posibilidades que ofrece cada uno de estos servicios a la hora de reutilizar la información. A raíz de este objetivo genérico surge otro más concreto:
- Analizar el sistema de recomendación musical **Last.fm** (<http://www.last.fm/>), y a los servicios que ofrece a través de su API.
- El estudio de diversas **herramientas de creación de mashups**. Análisis de sus posibilidades y carencias a través de la experimentación. Dentro de este objetivo genérico surge otro más concreto:
- El estudio de la herramienta de creación de mashups **Yahoo! Pipes**.. No requiere conocimientos de programación, y permite crear aplicaciones de forma rápida y sencilla. Actualmente se encuentra en fase beta, lo que indica que no está completamente terminada, y por tanto puede contener fallos. Intentaremos analizar las ventajas y desventajas de esta aplicación, así como identificar posibles errores que pueda arrojar al encontrarse todavía en fase beta. Para ello se crearán una serie de ‘pipes’ que utilicen la información ofrecida por el sistema de recomendación musical Last.fm, así como de otras aplicaciones, y que ofrezcan unos nuevos servicios basados en las fuentes de información primaria.
- Partiendo de la agenda de conciertos de la web Manerasdevivir (<http://www.manerasdevivir.com/conciertos.php>), que ofrece un listado de conciertos de música nacional en una web estática con contenido legible para el ser humano pero no para una máquina; se establece el objetivo de extraer el contenido utilizando la técnica de **screen scraping**, consistente en analizar el código de la web estática para extraer el contenido y conferirle la cualidad de ser reutilizable y legible por una máquina.
- Investigación de las posibilidades que ofrece la aplicación **Songbird**, que combina un reproductor musical con un navegador Web. En base a las posibilidades que ofrezca, utilizarlo como elemento de unión para todos los componentes que hayamos creado con anterioridad.

## 2. Estado del arte

En el apartado de estado del arte, empezaremos hablando de qué es un mashup, continuaremos analizando distintos servicios web orientados a la música que encontramos en Internet, analizaremos algunas fuentes de información que se pueden encontrar en Internet relacionadas con la música, directa o indirectamente, hablaremos de algunas tecnologías que o bien están relacionadas con la música o pueden ser herramientas interesantes para la creación de nuestro mashup, explicaremos la técnica de screen scraping, y hablaremos sobre los servicios web.

## 2.a. Integración de aplicaciones heterogéneas (mashups)

---

Un “mashup” es un tipo de aplicación web que combina el contenido de dos o más aplicaciones web, para crear un nuevo contenido completo.

El término “mashup” fue extraído del mundo de la música, en el cual se utiliza para referirse a un nuevo tema creado a partir de otros ya existentes. Al igual que en la música, en el mundo de Internet un *mashup* es una forma innovadora de componer contenidos (a menudo utilizando orígenes de datos diferentes).

### 2.a.i. Tipos de mashups atendiendo al contenido

Existen distintos tipos de mashups, con diferentes formas de presentar la información, y también con diferentes fines. A continuación se describen los géneros más populares.

#### Mapas

En la actualidad, se almacenan grandes cantidades de información relacionada con localizaciones. Esta información se presta de forma especial para ser presentada gráficamente en un mapa. Este hecho, unido a la existencia de aplicaciones web geográficas, como Google Maps o Yahoo! Maps, y sus interfaces públicas (APIs), hacen posible la creación de Mashups que permiten relacionar datos con sus localizaciones geográficas de forma gráfica.

#### Vídeo y foto

En los últimos años han ido apareciendo aplicaciones web que permiten el hospedaje de fotos y vídeos, como es el caso de Flickr y Youtube; aplicaciones que además permiten agregar datos semánticos a los ficheros almacenados, como el autor del fichero, el contenido de éste, dónde y cuando fue tomado el video/foto, etc. Gracias a la existencia de estas aplicaciones, y a la posibilidad de añadir datos semánticos, es posible mezclar fotos o vídeos con otra información, asociándolos a través de los metadatos, que son los que añaden contenido semántico. Por ejemplo, se puede analizar la letra de una canción y mostrar imágenes relacionadas con dicha letra, o en base a una noticia de la web de un periódico, mostrar vídeos relacionados con ésta.

#### Búsquedas y compras

Otra forma popular de mashups es la de crear buscadores/comparadores de precios entre artículos de dos o más webs de venta por Internet, por ejemplo Amazon y Ebay, permitiendo filtrar por precio, etc.

## Noticias

Desde hace años, las webs de noticias y periódicos en línea vienen utilizando sistemas de sindicación de contenido, como RSS o Atom. Gracias a esto, es posible crear un mashup de noticias personalizadas, mezclando por ejemplo noticias de distintos medios relacionadas con los tópicos que le interesan al usuario.

### *2.a.ii. Tipos de mashup atendiendo a su localización*

Generalmente los mashup se almacenan y ejecutan en una misma localización. Ésta puede variar dependiendo de los requisitos, y las alternativas que existen son: mashups ejecutados en servidor, mashups ejecutados en cliente, o mashups ejecutados en cliente y servidor.

#### Mashup ejecutados en servidor

En este caso, el contenido se genera de forma dinámica en el servidor, usando tecnologías de generación de contenidos como Java, CGI, PHP o ASP. Una vez generado el contenido, éste es enviado al navegador del cliente, el cual se encarga de mostrarlos.

Ventajas:

- Menor consumo de recursos en la máquina de cliente.
- Independencia del navegador del cliente. Al ejecutarse en el servidor, no requiere de una tecnología específica en la máquina del cliente.

Inconvenientes.

- Dependencia de un servicio de terceros. Al estar almacenado en un servidor de terceros, pueden producirse problemas como indisponibilidad, errores (p.e. timeout), etc.

#### Mashup ejecutados en cliente

El contenido puede también ser generado de forma dinámica en la máquina del cliente, en el navegador a través de un script (como Javascript) o applets, o en una aplicación independiente. El cliente se encarga de solicitar la información inicial a las fuentes primarias y generar el contenido final.

Ventajas:

- Independencia de un servidor de terceros para la generación del mashup, excepto en el caso de las fuentes primarias.

- Generalmente, menor tiempo de generación del mashup, ya que tenemos una máquina dedicada al efecto.

Inconvenientes:

- Mayor volumen de información a descargar. En lugar de tener que descargar el contenido final, es necesario descargar toda la información de las fuentes primarias.

## Mashup ejecutados en servidor y cliente

En ocasiones, debido a ciertos requisitos, o dado el carácter heterogéneo de las fuentes primarias, puede producirse la necesidad de tener que generar un mashup ‘mixto’, que se ejecute parte en la máquina de cliente, y parte en la máquina de servidor.

## Comparativa

A continuación se muestra una tabla comparativa entre mashup ejecutados en servidor, y mashups ejecutados en cliente. Dentro de los mashups ejecutados en cliente, podemos diferenciar entre mashups ejecutados en el navegador, y mashup ejecutados en aplicaciones independientes o ‘widgets’.

	Servidor	Cliente	
		Navegador	Aplicación indep.
Tiempo de generación	Alto	Bajo	Bajo
Volumen de info. a recibir	Bajo	Alto	Alto
Dependencia de plataformas	Bajo	Medio	Alto

Observamos como los mashup ejecutados en servidor son independientes de la plataforma de cliente. Los ejecutados en el navegador de cliente, tienen una cierta dependencia, ya que suele ser necesario la instalación de applets, extensiones, etc . en el navegador. Por último, los ejecutados en una aplicación independiente suelen ser los más dependientes de la plataforma, dado que comúnmente se trata de aplicaciones compiladas para una aplicación específica.

## 2.a.iii. Tecnología

Los mashup utilizan técnicas variadas para ofrecer el contenido final. A continuación se comentan las tecnologías más comunes:

## La arquitectura

Un mashup está compuesto por tres piezas fundamentales física y lógicamente separadas:

- **Un proveedor de servicio / API.** Son los proveedores del contenido, muchas veces de forma involuntaria. Algunos proveedores de servicio, para facilitar la tarea de extraer la información, ofrecen su contenido a través de interfaces públicas (APIs) como REST, o sindicación de contenidos como RSS o Atom. Aún así, hay otros que no lo hacen, pero sí permiten que el contenido sea reutilizado. En estos casos, es necesario hacer “screen scraping”, una técnica que permite extraer el contenido de un proveedor mediante una herramienta que trata de analizar las páginas webs y separar el contenido del diseño, para extraer el primero. Estas páginas web, se puede decir que únicamente fueron diseñadas para consumo humano.
- **El sitio que almacena el mashup.** Es donde el mashup se alberga, y en ocasiones, también donde se ejecuta. Los mashups pueden ejecutarse tanto en la parte del servidor como en la parte del cliente.
  - Mashup ejecutados en servidor.
  - Mashup ejecutados en cliente.
  - Mashup ejecutados en servidor y cliente.
- **El navegador de cliente.** Aquí es donde se renderiza gráficamente la aplicación y donde toma lugar la interacción con el usuario. Como se ha comentado anteriormente, hay ocasiones en los mashups son ejecutados en el navegador del cliente.

## AJAX

Ajax es el acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML). Más que una tecnología específica, Ajax es un modelo de aplicación web. Se compone de varias tecnologías orientadas a la carga asíncrona de contenido y su presentación:

- XHTML y CSS para la presentación.
- DOM (Document Object Model), que permite la modificación dinámica del contenido del documento.
- Intercambio asíncrono de datos, generalmente mediante XML.
- Scripts de usuario, generalmente JavaScript.

La utilización de estas tecnologías en conjunto tiene como objetivo crear una experiencia de navegación cómoda para el usuario, intercambiando pequeñas cantidades de datos con los servidores de contenidos, una técnica mucho más rápida y refinada que recargar la página y renderizarla de nuevo tras cada acción del usuario.

Esto se consigue mediante una técnica de comunicación asíncrona con el servidor en segundo plano, de manera que se pueden realizar peticiones de información mientras el usuario continúa interactuando con la aplicación.

Con esto se consiguen una serie de ventajas, como son la interactividad, velocidad y usabilidad, que convierten a AJAX en una técnica muy interesante para los creadores de mashups dadas las necesidades específicas de este tipo de aplicaciones como la velocidad a la hora de solicitar nueva información, la interactividad con el usuario, etc.

## Protocolos de comunicaciones

Existen varias alternativas, pero las dos más conocidas y utilizadas son SOAP y REST. Ambos son protocolos independientes de la plataforma o el lenguaje de programación, y su objetivo es la comunicación entre servicios remotos.

Se habla más en profundidad de los protocolos de comunicaciones en el apartado 2.f.

## Screen scraping

Cuando el proveedor de un servicio web carece de una API para acceder a los contenidos de su aplicación web, el desarrollador del mashup se ve obligado a utilizar la técnica de *screen scraping* o simplemente *scraping*, con el fin de extraer la información que desea mezclar.

Se habla más en profundidad del scraping en el apartado 2.d.

## Web semántica y RDF

La necesidad de utilizar *screen scrapers* está directamente relacionada con el hecho de que el contenido creado para consumo humano, generalmente no está indicado para el consumo de máquinas. Aquí es donde entra en juego el término Web Semántica, que se identifica con la visión de que la web actual, cuyo contenido, sólo legible por humanos, puede suplementarse añadiendo información legible por máquinas, o lo que es lo mismo, añadiendo contenido semántico. La Web Semántica tiene el objetivo de crear una infraestructura Web que añade metadatos a los datos, para darles así significado, haciendo posible su reutilización, automatización e integración.

La familia de especificaciones W3C, conocida como Resource Description Framework (**RDF**), sirve a este propósito de proveer metodologías para **establecer estructuras semánticas** que describen datos. XML por sí mismo no es suficiente, ya que es tan arbitrario que existen varias formas de codificar un mismo conjunto de datos.

**RDF-Schema** o *Esquema RDF* es una extensión semántica de RDF que añade a éste la habilidad de codificar conceptos de forma que sean legibles por máquinas. Una vez que los objetos de datos se pueden describir siguiendo un modelo de datos, RDF añade la posibilidad de **construir relaciones** entre ellos utilizando la tripla sujeto-predicado-objeto (el sujeto S tiene una relación R con el objeto O). La combinación de un modelo de datos y un sistema de relaciones permite la creación de **ontologías**, que son estructuras jerárquicas de conocimiento, sobre las que se pueden realizar búsquedas, y **realizar razonamientos formales** sobre ellas.



## Redifusión web

La redifusión web o “sindicación” permite que un sitio web que desea distribuir su contenido pueda hacerlo, a través de diferentes mecanismos.

Generalmente, la redifusión web consiste en crear un web feed o fuente web, disponible desde un sitio web con el fin de proveer a otra gente de un sumario del contenido más reciente del sitio web (normalmente las últimas noticias o publicaciones).

El término redifusión web también puede referirse para describir algunos tipos de licencias de contenidos que permiten la reutilización del contenido.

Los dos formatos más conocidos de redifusión web son RSS y Atom.

- RSS es una familia de formatos de redifusión basados en XML. Su acrónimo se utiliza para referirse a los siguientes estándares:
  - Rich Site Summary (RSS 0.91)
  - RDF Site Summary (RSS 0.9 y 1.0)
  - Really Simple Syndication (RSS 2.0)
- Atom es un protocolo de redifusión similar. Es un estándar del IETF<sup>1</sup>, que trata de mejorar algunas de las características de RSS.

---

<sup>1</sup> Internet Engineering Task Force, en castellano Grupo de Trabajo en Ingeniería de Internet, es una organización de normalización .

## 2.b. Servicios Web

---

Un Servicio Web es un sistema que permite la interoperatividad extremo a extremo entre dos máquinas a través de una red. Está formado por un conjunto de protocolos y estándares abiertos, que definen una interfaz a través de la cual se comunican dos o más máquinas, independientemente del lenguaje de programación en que estén diseñadas o la plataforma sobre la que se ejecuten.

La simpleza de los protocolos utilizados en la Web, como HTTP, les confiere una gran potencia y versatilidad. Lo mismo ocurre con los navegadores de cliente, componentes muy genéricos que, aunque pueden ampliar sus capacidades mediante applets, no dejan de ser genéricos. Es aquí donde los servicios web encuentran su sitio, buscando un modelo de aplicación específica de cliente que interactúa con un servicio que ofrece una interfaz orientada a una funcionalidad específica sobre Internet.

Los servicios web son un conjunto de protocolos que pueden ser usados de distintas formas. Las más comunes son RPC, SOAP y REST.

### 2.b.i. RPC

RPC, acrónimo de *Remote Procedure Call* (Llamada a Procedimientos Remotos en español) es una tecnología que **permite ejecutar una subrutina o procedimiento en otra máquina a través de una red**. Si el software en cuestión está escrito utilizando el paradigma de la orientación a objetos, a esta operación se le llama invocación remota.

RPC es el método más antiguo de invocación de métodos remotos, y sigue el modelo **cliente-servidor**. La comunicación es iniciada por el cliente, que envía un mensaje de petición a un servidor remoto, con el fin de ejecutar un procedimiento específico usando ciertos parámetros. El cliente permanecerá bloqueado hasta recibir una respuesta del servidor.

Una importante diferencia entre RPCs e invocaciones locales es que una invocación remota puede fallar por diversos motivos relacionados con la comunicación. Esto puede dar lugar a problemas cuando la aplicación cliente invoca un determinado procedimiento remoto, y se crea la incertidumbre de si el método remoto fue realmente invocado.

Con el fin de permitir la comunicación con el servidor a diferentes clientes independientemente del lenguaje en el que han sido escritos, es común utilizar un **Lenguaje de Descripción del Interface** (Interface Description Language IDL), que permite que distintas plataformas puedan invocar los métodos remotos del servidor.

Existen distintas implementaciones de RPC, como:

- RPC de Sun, también llamada ONC RPC. Originalmente desarrollado por Sun Microsystems, como parte de su sistema de ficheros NFS.
- DCE/RPC, Distributed Computing Environment. Permite al software funcionar sobre múltiples ordenadores, simulando que el conjunto fueran uno sólo.

## *2.b.ii. SOAP*

SOAP acrónimo de Simple Object Access Protocol, es un protocolo para intercambiar mensajes XML sobre redes de computadoras descentralizadas y distribuidas. Ha sido desarrollado para ser independiente de cualquier modelo de programación y de otras semánticas. Forma parte de la pila de protocolos de los servicios web, que se compone de:

- **Protocolo de transporte**, responsable de transportar mensajes sobre una red. Algunos ejemplos son HTTP, FTP o SMTP.
- **Protocolo de mensajería**, responsable de codificar los mensajes en formato XML de forma que sea legible tanto para la máquina origen como para la máquina destino. Dentro de este protocolo se enmarca **SOAP**.
- **Protocolo de descripción**, encargado de describir la interfaz de un determinado servicio web. El más usado es WSDL (Web Services Description Language).
- **Protocolo de descubrimiento**, que centraliza los servicios en un registro común, con su localización y descripción, de forma que se simplifica la tarea de descubrir qué servicios están disponibles en la red. El más común es UDDI (Universal Description, Discovery and Integration).

En la mayoría de los casos se escoge HTTP como protocolo de transporte, al trabajar bien en la infraestructura de Internet, ya que es un protocolo que suele pasar los filtros de los firewall, al contrario que otros protocolos como DCOM.

Hay dos componentes clave en la especificación de SOAP:

- El uso de un **formato de mensaje XML**, para una codificación independiente de la plataforma. Se utiliza XML como formato de mensaje por su amplio uso y por su condición de estándar y abierto. XML no es el lenguaje más ligero para ser transportado a través de una red, pero tiene grandes beneficios, como que es legible por humanos, al contrario que otros formatos, que al ser binarios tienen un menor tamaño pero no son legibles por humanos.
- La **estructura del mensaje**, consistente en **una cabecera y un cuerpo**. La cabecera se utiliza para intercambiar información contextual que no es específica para el contenido del mensaje, por ejemplo información de autenticación. El cuerpo del mensaje encapsula el contenido específico de la aplicación.

A continuación se comentan las ventajas y desventajas de este protocolo:

- **Ventajas:**

- Al ser usado sobre HTTP permite una comunicación sencilla a través de proxies y firewalls.
- Su versatilidad le permite funcionar sobre diferentes protocolos de transporte además de HTTP, como por ejemplo SMTP.
- Es independiente de cualquier plataforma.
- Es independiente de cualquier lenguaje de programación.
- Es simple y extensible.

- **Desventajas:**

- Al utilizar XML, SOAP puede ser considerablemente más lento en relación a otros protocolos como CORBA. Esto no es un inconveniente para mensajes pequeños, pero sí cuando crece el tamaño del mensaje.
- Cuando se utiliza sobre HTTP, los roles de los extremos son fijos, de forma que sólo el cliente puede acceder a los servicios del servidor y no viceversa.

## *2.b.iii. REST*

REpresentational State Transfer, en castellano Transferencia de Estado Representacional es una técnica de arquitectura software para sistemas hipertexto distribuidos como la World Wide Web. El término se originó en el año 2000 en una tesis doctoral sobre la web escrita por Roy Fielding. Actualmente el término REST se utiliza para describir cualquier interfaz web simple que utiliza XML y HTTP, sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes como SOAP.

Los sistemas que siguen los principios REST se llaman con frecuencia RESTful. REST se basa en una serie de principios, los mismos que han permitido que La Web sea tan escalable:

- **Un protocolo cliente/servidor sin estado.** Cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión (algunas de estas prácticas, como la reescritura de URL, no son permitidas por REST).
- **Un conjunto de operaciones bien definidas** que se aplican a todos los recursos de información: HTTP define un conjunto pequeño de operaciones, las más importantes son POST, GET, PUT y DELETE.
- **Una sintaxis universal** para identificar recursos. En un sistema REST cada recurso es direccionable únicamente a través de su URI.

- **El uso de hipermedios** tanto para la información de la aplicación como para las transiciones de estado de la aplicación: la representación de este estado en un sistema REST son típicamente HTML o XML. Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional.

## Recursos

Un concepto importante en REST es la existencia de **recursos**, cada uno de los cuales puede ser accedido usando un identificador global (URI, Universal Resource Identifier). Como resultado, el servidor devolverá una **representación del recurso**, no el recurso en sí. Por ejemplo, si el recurso es un círculo el servidor podría responder con un fichero vectorial en formato SVG que representa el círculo indicando el punto central y un radio.

La World Wide Web es un buen ejemplo de diseño RESTful. En su mayor parte se ajusta a los principios REST. La Web consta de:

- Hypertext Transfer Protocol (HTTP).
- Tipos de contenido como Hypertext Markup Language (HTML).
- Soporta código bajo demanda a través de JavaScript y otros lenguajes.
- Tiene una interfaz uniforme para acceder a los recursos, que se compone de URIs, métodos, códigos de estado, cabeceras y contenidos distinguidos por tipos MIME.

## 2.c. Servicios web orientados a la música

En los últimos años han ido surgiendo una serie de servicios web orientados a la música siguiendo en general dos vertientes principales en lo referente al objetivo:

- Ofrecer recomendaciones musicales basándose en el conocimiento sobre los gustos del usuario.
- Ofrecer un servicio para dar a conocer nuevos artistas y ser campo base para la obra de artistas ya renombrados.

A continuación comentamos algunos de los más importantes servicios web orientados a la música.

### 2.c.i. Last.fm

**Last.fm** es un sistema de recomendaciones musicales y de radio a través de Internet. Utiliza una base de conocimiento y sistema de recomendación llamado Audioscrobbler, que va almacenando datos sobre los hábitos de escucha de los usuarios de Last.fm.

El **perfil** de cada usuario se construye en base a la música que escucha en sus reproductores habituales (existen extensiones que recopilan esta información), así como a la música que escucha en el reproductor de radio de Last.fm. El usuario tiene la opción de marcar una canción como favorita o vetarla en caso de que no le guste.

Basado en los datos almacenados, Last.fm ofrece una **radio personalizada** a través de Internet, permitiendo al usuario escuchar música acorde con sus gustos, elegir la emisora en base a un género musical, o por similitud a un grupo/artista concreto.

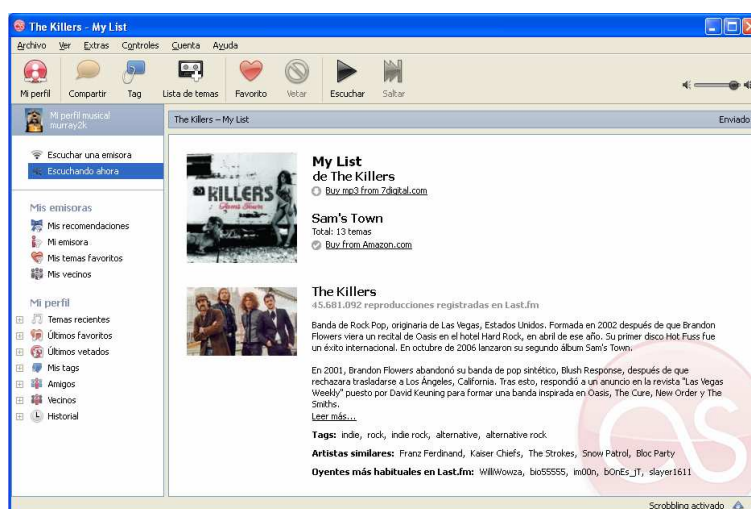


Ilustración 1. Radio de Last.fm

Además de ofrecer un servicio de radio, Last.fm ofrece **recomendaciones musicales** basándose en las relaciones que a través de la estadística establece entre artistas, permitiendo al usuario conocer nuevos artistas y géneros.

Al añadir la función de **etiquetado** tanto a artistas como a temas y álbumes, las posibilidades de personalización se multiplican ya que dicho tagging o etiquetado permite la clasificación de la música en múltiples formas: por estilos (rock, r&b, etc), estados de ánimo (happy, tranquilo, etc.).

El usuario también tiene la opción de conocer sus propios **hábitos de escucha**, así como los de otros usuarios, de forma que puede obtener de forma gráfica listados de los artistas o temas más escuchados, usuarios con gustos comunes (vecinos), e incluso puede crear sus propias listas de reproducción.

A su vez, se pueden crear **redes sociales de amigos**, grupos de escuchas comunes sobre los que verter comentarios y abrir hilos de discusión. Un usuario puede recomendar un artista o un tema a otro usuario, amigo o vecino.

El sistema también ofrece una **agenda de conciertos** por artista, que puede ser editada por los usuarios, de forma que si un usuario conoce una fecha de actuación puede darla de alta en el sistema, así como crear o añadir datos acerca de la ubicación del evento.

## Audioscrobber

**Audioscrobber** recopila datos sobre los hábitos de escucha mediante extensiones que el usuario integra en su reproductor habitual, así como de las escuchas que el usuario hace a través de la radio personalizada.

El sistema nació de forma independiente a Last.fm, como **proyecto de fin de carrera** de un estudiante de informática de Inglaterra, que desarrollo las primeras extensiones que permitían enviar al sistema los datos recopilados. Más tarde, Last.fm empezó a utilizar los datos recopilados por Audioscrobber.

## API

Además de servir como fuente de información para el sistema Last.fm, Audioscrobber ofrece una serie de datos a través de su **API**, en fase beta y libre para uso no comercial. Algunos de estos datos con:

- Datos del **perfil de usuario**, como temas, artistas o álbumes preferidos; escuchas recientes, etc.
- Datos sobre **artistas**, como artistas similares, temas y álbumes más escuchados, conciertos cercanos, etc.
- Datos sobre **temas**, como temas similares, usuarios que más lo escuchan, o etiquetas más usadas.
- Datos sobre **etiquetas**, como artistas, temas o álbumes que más se ciñen a esa etiqueta, o las etiquetas más populares.

- Datos **geográficos**, como los artistas o los temas que más se escuchan en un país.

Dicha API sigue la arquitectura **REST**.

Además, Audioscrobbler, integra el uso de **microformatos** que permiten añadir contenido semántico a los datos almacenados.

## *2.c.ii. Foafing the music*

Foafing the music es un sistema de recomendación musical basado en un perfil del usuario y en sus hábitos de escucha. En función de factores como gustos, hábitos, localización geográfica, el sistema ofrece recomendaciones musicales personalizadas.

El sistema se basa en dos orígenes de datos para ofrecer dichas recomendaciones:

- Un perfil FOAF (Friend of a Friend). Es una ontología que describe personas, sus actividades y sus relaciones con otras personas y objetos. Para definir esta ontología, se utiliza el lenguaje de modelado RDF (Resource Description Framework). Dado que parte de esta ontología se utiliza para reflejar los gustos del usuario, se aprovecha esta funcionalidad para analizar sus gustos musicales.
- Los hábitos de escucha del usuario, extraídos de la base de datos Audioscrobbler (comentado anteriormente en el apartado 2.a.i. Last.fm).

Foafing the music ofrece servicios relacionados con:

- **Artistas.** Basado en el perfil FOAF del usuario, el sistema es capaz de ofrecer, en base a los artistas preferidos del usuario y artistas relacionados:
  - Un listado de artistas relacionados, incluyendo un ratio de relación entre ellos.
  - Un listado de novedades (discos, sencillos, DVDs, etc.). Incluye enlaces a tiendas en línea como iTunes o Amazon.
  - Un listado de reseñas de diferentes medios, como [www.rateyourmusic.com](http://www.rateyourmusic.com) o [www.rollingstone.com](http://www.rollingstone.com).
  - Un listado de enlaces a mp3-blogs, con reseñas de temas y discos, así como enlaces torrent y descarga directa de música.
  - Un listado de podcasts.
- **Música.** Basado en los hábitos de escucha almacenados en Audioscrobbler, Foafing the music ofrece información estadística sobre los hábitos de escucha, como por ejemplo los cuatro artistas más escuchados últimamente.
- **Listas de reproducción.** A través del servicio Webjay, Foafing the music ofrece acceso a las listas de reproducción que el usuario haya creado en este sistema.



### *2.c.iii. MyStrands*

MyStrands es un sistema social de recomendaciones de productos en tiempo real a través del PC, el teléfono móvil y otros dispositivos con conexión a Internet.

Algunos de los productos a los que MyStrands está orientado son los siguientes:

- **MoneyStrands.** Este servicio permite visualizar de forma centralizada toda la información financiera que concierne al usuario. Además, ofrece recomendaciones personalizadas de productos financieros y servicios.
- **Recomendaciones sociales para comercio electrónico.** Es un motor de recomendación que se instala como una extensión en los comercios web, y que permite añadir funcionalidades de recomendaciones a éstos.
- **Comunidad musical y recomendaciones.** Permite al usuario encontrar otros usuarios con gustos similares, conocer qué escuchan los demás usuarios en un momento dado, y contactar con ellos. Ofrece recomendaciones musicales en base a los gustos del usuario, todo ello gracias a una aplicación desarrollada para Windows, Mac y Linux.
- **MyStrands TV.** Utilizando la API de Youtube, MyStrands TV es capaz de ofrecer al usuario, en base a un artista determinado, videos de éste y otros artistas relacionados.
- **OpenStrands** es el nombre que se le da a la API de MyStrands, gratuita para usuarios registrados.
- **PartyStrands.** Es un servicio interactivo para organizadores de fiestas, bares, y DJs, que permite la interacción del público. Hablaremos más ampliamente de este servicio en el siguiente subapartado.

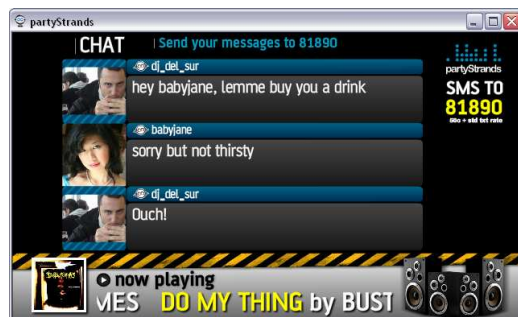
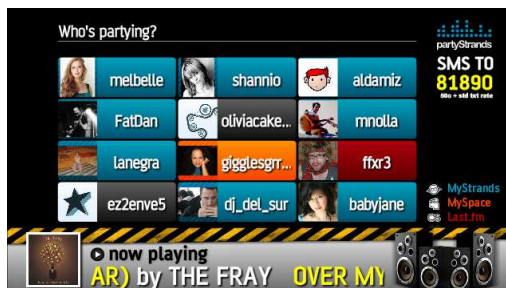
### **PartyStrands**

PartyStrands es un servicio interactivo para organizadores de fiestas, que permite la interacción entre asistentes, organizadores, y el contenido multimedia que el local ofrece.

Básicamente, permite a los asistentes, mediante su teléfono móvil, elegir el tipo de música que suena, así como escoger un tema determinado para que suene. Además, ofrece otras características, como:

- Enviar imágenes y mensajes de texto desde el teléfono móvil para que sean mostradas en la pantalla.
- Organizar votos sobre canciones, bebidas, asistente más popular, etc. Los asistentes pueden elegir al ganador desde su teléfono móvil.
- Existe la posibilidad de mostrar en la pantalla el nombre de artista y tema que está sonando en cualquier momento.

- Si el organizador tiene usuario de MyStrands la música que sonará será la que sea afín a sus gustos según su perfil de MyStrands.
- Al día siguiente es posible recordar la música que sonó, ya que se habilita un sitio web para cada nueva ubicación de una fiesta en la que los usuarios pueden ver los mensajes de texto, fotos, asistentes, y lista de reproducción que sonó el día anterior en la fiesta. Toda esta información se actualiza en tiempo real.



## API

La API de MyStrands se llama OpenStrands, es libre para uso no comercial, sigue la arquitectura REST y, ofrece servicios como:

- Opciones de cuenta de usuario, como validación, actualización de correo electrónico, actualización de contraseña, cambio del perfil de usuario, actualización del alias, consulta de un perfil de usuario, etc.
- Servicios de catálogo, retornan información en base a un artista, álbum o tema,
- Listados de géneros, de artistas por género, de álbumes, etc.
- Álbumes y temas más recientes, los que han sido lanzados recientemente.
- Recomendaciones de artistas, temas y álbumes.
- Descarga de listas de reproducción.
- Búsquedas de usuarios, etiquetas.
- Estadísticas de escucha, como el artista del día, de la semana, etc.

## 2.c.iv. Google Music Trends

This week's top songs			
		All Genres	All Countries
Rank	Title	Artist	Change
1	<a href="#">What I've Done</a>	<a href="#">Linkin Park</a>	↑ +1
2	<a href="#">In the End</a>	<a href="#">Linkin Park</a>	New!
3	<a href="#">Turn Se Hi - Web Search</a>	<a href="#">Mohit Chouhan - Web Search</a>	...
4	<a href="#">Numb</a>	<a href="#">Linkin Park</a>	...
5	<a href="#">Bleed It Out</a>	<a href="#">Linkin Park</a>	...
6	<a href="#">Hey There Delilah</a>	<a href="#">Plain White T's</a>	↑ +2
7	<a href="#">Boulevard Of Broken Dreams</a>	<a href="#">Green Day</a>	↑ +2
8	<a href="#">No One</a>	<a href="#">Alicia Keys</a>	↑ +7
9	<a href="#">How To Save A Life</a>	<a href="#">The Fray</a>	↑ +10
10	<a href="#">You're Beautiful</a>	<a href="#">James Blunt</a>	↑ +1
11	<a href="#">Shadow Of The Day</a>	<a href="#">Linkin Park</a>	↓ -4
12	<a href="#">Don't Matter</a>	<a href="#">Akon</a>	↓ -6
13	<a href="#">When You're Gone</a>	<a href="#">Avril Lavigne</a>	New!
14	<a href="#">Say It Right</a>	<a href="#">Nelly Furtado</a>	...
15	<a href="#">Some Unholy War</a>	<a href="#">Amy Winehouse</a>	New!
16	<a href="#">Girlfriend</a>	<a href="#">Avril Lavigne</a>	New!
17	<a href="#">Somebody's Me</a>	<a href="#">Enrique Iglesias</a>	New!
18	<a href="#">Paranoid Android</a>	<a href="#">Radiohead</a>	New!
19	<a href="#">Ye Ishq Hai - Web Search</a>	<a href="#">Shreya Ghosal</a>	↓ -1
20	<a href="#">The Reason</a>	<a href="#">Hoobastank</a>	↓ -8

Music Trends es un servicio de Google que muestra estadísticas semanales sobre las canciones más escuchadas por los usuarios del servicio Google Talk (mensajería instantánea y voz sobre IP) que tengan activada la opción de “*compartir tu estado musical*”. Dichos datos, actualmente pueden ser filtrados por género musical y/o por país de escucha, aunque esta última opción es algo limitada, ya que actualmente sólo hay 9 países para elegir.

Activando la opción ya mencionada de “*compartir tu estado musical*”, el usuario obtiene la posibilidad de acceder a un historial de escuchas, el cual permite filtrar entre todas las canciones que ha escuchado.

En la mayoría de los casos, las canciones obtenidas como resultado tienen un enlace a sus respectivas entradas en **Music Search de Google**, un servicio de búsqueda orientado a la música de Google. En este servicio podemos encontrar datos relacionados con:

- **Grupos:** discografía, noticias, fotos, etc.
- **Discos:** valoraciones de los usuarios, listado de pistas, y enlaces a tiendas donde poder adquirirlo.
- **Canciones:** datos sobre la canción, versiones, letras de la canción, y enlaces a tiendas donde poder adquirirlas.

Ofrece filtrado de resultados a través de la URL, siguiendo el formato:

**[http://www.google.com/trends/music?genre=<género>&where=<cod\\_país>](http://www.google.com/trends/music?genre=<género>&where=<cod_país>)**

Sin embargo, los resultados no se ofrecen en un formato reutilizable, sino que son mostrados en pantalla.

## 2.c.v. MySpace

MySpace es un sitio web orientado a la interacción social, que ofrece servicios como creación de perfiles personales, redes de amigos, grupos, alojamiento de fotos, vídeos y música, blogs, etc.

El objetivo de la creación de perfiles personales es el darse a conocer, si bien pueden existir diferentes motivaciones, como dar a conocer la obra y biografía en caso del artista, o darse a conocer de forma personal, dado que es una potente herramienta de comunicación social.

Algunas características importantes para los músicos son:

- Al crear un perfil existe la posibilidad de “colgar” hasta seis canciones en formato mp3 para dar a conocer la obra del artista. Al acceder al perfil del artista, el usuario encontrará un reproductor web en el que poder escuchar la obra de este.
- Posibilidad de colgar fotos y enlazar vídeos de otras webs, como Youtube.
- Creación de blogs.
- Creación de una red social de amigos, que permite conocer otros músicos, así como darse a conocer a través de aficionados que también son usuarios del servicio. Existe la posibilidad de mostrar una lista de amigos en el sitio MySpace del artista.
- Hilos de comentarios.
- Mensajería instantánea.

En los últimos tiempos, MySpace se ha convertido en una potente herramienta para los artistas emergentes, ya que, potenciado por su creciente red social permite a los artistas darse a conocer de forma rápida. Además, gracias a las herramientas comentadas, es una interesante plataforma para publicar parte de la obra del artista, de forma que ya no se hace tan imprescindible el que el artista tenga que estar ligado a una discográfica para darse a conocer o distribuir su obra.

Existen ejemplos de grupos que, sin ayuda de una discográfica, han logrado darse a conocer y tener una gran cantidad de seguidores gracias a este tipo de nuevas herramientas. Un ejemplo de esto es el grupo británico Arctic Monkeys.

## API

MySpace dispone de una API en formato REST, parte de la plataforma MDP (Myspace Developer Platform), que proporciona una interfaz para acceder a varios servicios de MySpace.

### *2.c.vi. Grooveshark*

Grooveshark es una red social de recomendación musical, con la singularidad de que la música no se almacena en un servidor sino que utiliza un motor P2P, de forma que todos los usuarios, a la vez que descubren música, también la comparten. Actualmente está en fase beta.

El usuario tiene la opción de escuchar cualquier canción del sistema mediante el reproductor web integrado, puede crear listas de reproducción personalizadas y tiene la opción de descargar cualquier canción por el precio de 0,99 dólares.

Antes de empezar a utilizar Grooveshark es necesario descargar e instalar una aplicación, que será la que hará de motor P2P para compartir ficheros musicales y poder recibir los que comparten los demás.

Algunos de los servicios destacados son:

- Perfiles de usuarios.
- Sistema de mensajes privados.
- Recomendaciones en función de nuestros gustos y música compartida.
- Listas de reproducción que se pueden compartir.

Grooveshark no dispone de ningún tipo de API.

## 2.d. Fuentes de información

---

### 2.d.i. Buscadores de música

De un tiempo a esta parte los buscadores web de mp3 han perdido popularidad debido a la masificación de los programas *peer to peer*, que hacen la tarea de búsqueda más sencilla, y simplifican sustancialmente la gestión de descargas. Sin embargo, hay algunos campos en los que los buscadores web siguen siendo mejores:

- Rapidez en las descargas sencillas. En el caso de descargar un tema individual, probablemente el método web sea más rápido debido a que se trata de una descarga directa y no es necesario esperar colas ni buscar fuentes como en el caso del P2P.
- Sencillez de implantación en aplicaciones dada la existencia de *APIs*.

Como siempre, sigue existiendo un vacío legal a la hora de compartir música sujeta a derechos de copyright. Sin embargo, los motores de búsqueda sostienen que sólo buscan enlaces a ficheros *MP3* hospedados por terceros. Las compañías discográficas ya no prestan tanta atención a los buscadores web como a los servicios *P2P*, a los que ahora consideran su principal amenaza.

Básicamente, los buscadores web de *MP3* se dedican a rastrear la web en busca de enlaces a ficheros *MP3*, para añadirlos a su base de conocimiento y más tarde puedan estar accesibles en las búsquedas de los usuarios.

Comentaremos algunos buscadores web de *MP3*, analizando los servicios que ofrecen, así como sus interfaces de aplicación (*API*).

#### Skreemr

Skreemr es uno de los buscadores de *MP3* más conocidos en la actualidad. Las búsquedas suelen ofrecer bastantes resultados, y su principal característica, que lo diferencia de los demás buscadores, es que asigna a cada resultado un sistema de puntuación del 1 al 10, al cual la denomina “**Audiorank**”, y que se construye en base a diversos factores, como:

- Confiabilidad del enlace.
- Calidad de sonido.
- Completitud del tema.
- Voto de los usuarios.

Los resultados se ordenan en base a este sistema de puntuación en orden descendente, de forma que los primeros resultados son los de mayor calidad.

Además del sistema “*Audiorank*”, Skreemr ofrece con cada resultado:

- La opción de reproducir el tema a través de su reproductor flash.
- La descarga directa.
- Acceso a tiendas en línea para adquirir el disco o el sencillo.
- Enlaces a diversas webs con contenidos relacionados, como letras de canciones, biografías, listado de conciertos, fotografías, vídeos, etc

El lado negativo es que, aunque el servicio tiene API, ésta sólo se puede utilizar si se tiene licencia, la cual no está accesible para estudiantes por el momento.

## Woonz

Woonz es otro buscador web de mp3, que ofrece gran cantidad de resultados de búsqueda, aunque éstos no son del todo precisos. Busca en otros formatos además de mp3, como por ejemplo wma.

Ofrece también un reproductor flash, que incorpora la posibilidad de añadir más de un tema a la lista de reproducción, y que además es incrustable mediante código HTML en cualquier web.

En este caso, el buscador no ofrece ningún tipo de API que permita acceder a los resultados desde otra aplicación.

## Seeqpod

Seeqpod es un potente buscador, no sólo de mp3, sino de vídeos (mayoritariamente en Youtube) y de documentos escritos relacionados con la búsqueda (mayoritariamente en Wikipedia). Está mayoritariamente construido en flash e integra un reproductor con lista de reproducción. Integra las siguientes funciones:

- Reproducción de ficheros de audio en su reproductor flash.
- Reproducción de ficheros de vídeo en su reproductor flash.
- Creación de listas de reproducción.
- Descarga directa de ficheros de audio.
- Enlace a documentos escritos relacionados con la búsqueda.

Como punto a su favor, incorpora una interfaz de aplicación (API) que sólo requiere haber creado un usuario en su sistema, acción que es gratuita. Utiliza la arquitectura REST, que permite pasar los parámetros de la petición a través de la URL, y devuelve los resultados en formato XML.

Además, la API incorpora un corrector ortográfico y recomendador de consulta, que ayuda a dar con el nombre de un artista/grupo en caso de no conocer el nombre exacto de éste.

## *2.d.ii. Wikipedia*

Wikipedia es una enciclopedia en línea, libre, colaborativa y multilingüe, que, basada en la filosofía “wiki”<sup>2</sup>, permite a cualquier usuario modificar los artículos que contiene con el fin de mejorar y crecer en contenido. Una filosofía cuestionada y criticada en un principio, pero que con el paso del tiempo ha demostrado ser el mejor mecanismo para convertir a Wikipedia en la enciclopedia más utilizada en el mundo.

Contiene artículos en más de 250 idiomas, alcanzando los 2,5 millones de artículos en inglés, y los 380.000 en español.

Al ser una enciclopedia colaborativa, la veracidad de la información es cuestionable, e incluso los administradores recomiendan que sea usada como fuente complementaria de información. Aún así, importantes medios como la revista “Nature” comparan ya Wikipedia en inglés con la Enciclopedia Británica en cuanto a veracidad de contenidos.

Vista la gran acogida y la rapidez de crecimiento que desde su creación en 2001 ha sufrido, las pretensiones de Wikipedia han cambiado de ser una enciclopedia colaborativa a pretender convertirse en la mayor base de conocimiento, abarcando por tanto muchos más campos que los que abarcaría una enciclopedia normal. Esto es posible gracias al dinamismo que aporta la colaboración de cualquier usuario, al ser posible hacer modificaciones instantáneas en su contenido, hecho que permite mantener actualizada la enciclopedia al momento.

Por este motivo, Wikipedia es una importante referencia en el sector de la música, ya que contiene biografías de gran cantidad de artistas, así como información sobre su discografía, fechas de conciertos, etc.

## **API**

La API de Wikipedia sigue la arquitectura REST, de forma que las consultas se realizan a través de la URL, y los resultados son devueltos en diversos formatos, como JSON, XML, etc.

## *2.d.iii. Letras de canciones*

Existen multitud de webs que contienen letras de canciones, pero la mayoría de ellas comparten defectos:

---

<sup>2</sup> Sitio web en el cual el contenido de sus páginas puede ser modificado por el usuario a través del navegador.



- Lentitud.
- Publicidad.
- Imprecisiones en las letras.
- Discografías incompletas.
- Resultados en páginas estáticas, no reutilizables.

Sin embargo, existe una web llamada LyricWiki, que combina la funcionalidad de un “wiki” con un servicio de letras para canciones.

## LyricWiki

LyricWiki es un “wiki” con unas 670.000 páginas, el séptimo “wiki” más grande, según Wikimedia, a fecha 13 de junio de 2008. Contiene letras de canciones organizadas por artistas, álbumes, géneros, etc. Además utiliza el motor Google para ofrecer búsquedas internas.

Haciendo honor a su nombre, permite a cualquier usuario modificar las letras de una canción, así como añadir nuevos artistas, álbumes o temas en caso de no estar creados.

LyricWiki soluciona algunas de las deficiencias antes comentadas en las web de letras de canciones:

- No se experimenta una gran lentitud.
- La publicidad es escasa y muy localizada (los beneficios se destinan únicamente a subvencionar los costes de hosting).
- A ser modificable, las discografías suelen ser bastantes completas.
- Las letras son precisas ya que permite que cualquier usuario que vea un error lo corrija al instante.
- Los resultados son reutilizable ya que utiliza un interfaz SOAP.

Algunas de sus **características** son:

- Existen multitud de **extensiones** para reproductores multimedia que permiten obtener las letras de las canciones automáticamente cuando empiezan a sonar, todas estas extensiones hacen las peticiones a la API pública de LyricWki.
- Existe una **aplicación** para FaceBook que muestra letras de canciones de forma aleatoria, de forma que el usuario tiene que adivinar a qué canción pertenecen.
- Existe una **canción del día y un álbum de la semana**.

## API

LyricWiki ofrece una API pública, aún en desarrollo, accesible a través de dos arquitecturas:

- SOAP: el formato de salida es XML.
- REST: el desarrollo va un poco más atrasado que la interfaz SOAP. Ofrece la funcionalidad básica de obtener las letras de una canción. Ofrece distintos formatos de salida:
  - HTML (por defecto).
  - XML.
  - Texto.

Básicamente es la misma API pero con dos interfaces diferentes, aunque la interfaz REST aún está menos desarrollada que la interfaz SOAP.

## *2.d.iv. Medios*

Existen multitud de medios orientados a la música en Internet, como:

- Periódicos
- Blogs
- Revistas digitales
- Medios especializados
- Webs de crítica
- Páginas personales

A continuación comentamos algunas que o bien se han utilizado directa o indirectamente en el proyecto, o bien han sido candidatas; sin embargo, existe tal cantidad de webs orientadas a la música que sería imposible mencionarlas todas.

## **Manerasdevivir**

Manerasdevivir es un portal de música orientado al rock nacional. Contiene varios apartados:

- Un blog con actualidad sobre este tipo de música
- Una agenda de conciertos
- Un foro
- Una sección de entrevistas
- Una sección de lanzamientos
- Una sección de grupos con información de cada grupo y un enlace a su web oficial o MySpace.

- Fotos
- Descarga de música con licencia Copyleft<sup>3</sup>
- Vídeos
- Reportajes sobre conciertos y festivales.

Además, permite a cualquier usuario que visite la página la opción de añadir por sí mismo noticias o fechas de conciertos.

Durante los 11 años que lleva existiendo se ha ido convirtiendo en uno de los portales de referencia para la música rock nacional.

Todo el contenido del portal está protegido bajo una licencia “Creative Commons Atribución 3.0 Unported”, que permite copiar, distribuir, comunicar y ejecutar públicamente la obra, así como hacer obras derivadas bajo la condición de que es necesario reconocer y citar la obra.

La web no ofrece ningún tipo de API, si bien, el contenido de las webs puede ser extraído utilizando una técnica denominada “scraping”, mediante la cual se puede extraer el contenido de una web estática con el propósito de transformarlo en otro formato válido para su uso en otro contexto.

## Hipersónica

Hipersónica es un blog nacional sobre actualidad musical tanto nacional como internacional, generalmente de la escena alternativa.

El blog cuenta con nueve autores, que escriben sobre novedades musicales, conciertos, festivales, repastos discográficos de artistas, críticas musicales, etc.

Contiene además un top veinte de temas musicales actuales, actualizado cada semana a través de las votaciones de los visitantes.

El contenido del blog se ofrece a través de un feed RSS.

## Allmusic

Allmusic es una extensa guía sobre música, que contiene:

- Reseñas de discos y temas
- Biografías de artistas
- Discografías
- Relaciones entre artistas (de influencia, de seguidores, artistas similares, etc.)

---

<sup>3</sup> Copyleft es una forma de derechos de autor que elimina restricciones sobre la distribución y modificación de las obras con respecto al copyright, siempre que se mantengan los derechos que estableció el autor original de la obra.

Allmusic contiene una de las más grandes colecciones de música digital, con unos seis millones de temas digitalizados, así como la mayor biblioteca de carátulas, con medio millón de carátulas.

Cuenta además con 900 críticos musicales, siendo éste uno de sus puntos más fuertes, ya que en este terreno Allmusic es una de las referencias en los que a **crítica musical** se refiere en el mundo de Internet.

La web no ofrece ningún tipo de API, y su contenido está sujeto a derechos de copyright que impiden la reutilización de dicho contenido.

## Rolling Stone

Descendiente directo de la revista con el mismo nombre y con contenidos como música, cine y cultura popular, la web de Rolling Stone contiene artículos, reseñas, blogs, MP3, etc.

Desde el punto de vista musical, la web de Rolling Stone contiene un listado de artistas, artículos musicales, y reseñas de discos. Este último aspecto es el más interesante desde el punto de vista musical, siendo una de las referencias en Internet junto a la web de Allmusic.

Ofrece también listados como los 100 mejores artistas de todos los tiempos, los 500 mejores álbumes de todos los tiempos, o los 500 mejores temas de todos los tiempos; si bien los criterios de selección pueden no ser del todo objetivos.

La web de Rolling Stone advierte de que parte del contenido relacionado con los álbumes puede estar extraído de la web de Allmusic.

El contenido de la web [rollingstone.com](http://rollingstone.com) está sujeto a copyright, lo que implica que el contenido no puede ser reutilizado.

## *2.d.v. Buscadores de vídeos*

Existen innumerables buscadores de vídeo, casi todos son webs que permiten colgar, compartir y visionar vídeos en línea. Sin embargo, de entre estas webs existe una que siempre ha destacado frente al resto: Youtube.

### Youtube

Youtube fue creada en 2005 por tres trabajadores de PayPal, que decidieron utilizar el reproductor web de Adobe Flash para mostrar vídeos subidos por los usuarios, como vídeos caseros, clips de películas, clips de TV, vídeos musicales, videoblogging, etc. En 2006 Google adquiere la compañía por 1650 millones de dólares.

Actualmente es con diferencia la web de vídeos más utilizada en Internet. Youtube permite ver vídeos a los usuarios no registrados en el sistema, mientras que permite

colgar un número ilimitado de vídeos a los usuarios registrados. Existen ciertos tipos de contenidos que no están permitidos, como los de carácter ofensivo o sujeto a derechos de autor (cuando no sea el autor o alguien con su consentimiento el que coloque los vídeos).

Según datos de la compañía, más de 100 millones de vídeos son visionados diariamente. Según el medio Wall Street Journal, en agosto de 2006 Youtube albergaba alrededor de 6,1 millones de vídeos, y contenía medio millón de usuarios registrados. En abril de 2008 una búsqueda en la web de Youtube arroja 83,4 millones de vídeos y 3,75 millones de canales de usuario.

Uno de los contenidos más típicos es el de los vídeos musicales, así como grabaciones de conciertos, lo cual le confiere cierta trascendencia dentro de nuestros objetivos.

Youtube ofrece una API pública que sigue el modelo REST, accesible incluso para usuarios no registrados.

## *2.d.vi. Agendas de conciertos*

### **Last.fm**

Last.fm ya ha sido comentado en el apartado de servicios web orientados a la música. Sin embargo, además de ser un mero servicio de recomendación, ofrece también una completa agenda de conciertos, creada por los propios usuarios de la comunidad, que tienen la posibilidad de añadir y modificar datos sobre cada concierto y cada localización.

Todos los datos sobre conciertos en Last.fm están accesibles a través de su API pública y gratuita siguiendo el modelo de arquitectura REST. Además, Last.fm utiliza microformatos<sup>4</sup> en las consultas de eventos y de localizaciones de éstos.

### **Manerasdevivir**

Manerasdevivir es un portal de música rock nacional, sobre el que ya se ha hablado en el apartado de Medios. Sin embargo, se hará una pequeña reseña en este apartado de Agendas de conciertos, ya que esta web es una de las más importantes en lo referente a las agendas de conciertos de bandas nacionales, ofreciendo un listado con gran cantidad de artistas.

Se pueden filtrar los resultados por artista y/o por provincias. Sin embargo, los resultados son mostrados en una página estática que no ofrece ninguna forma de reutilización de contenido como un feed RSS o un fichero XML.

---

<sup>4</sup> Los microformatos son una forma de codificación de datos que permite añadirles significado semántico, para conferirles la condición de reutilizable. Así se consigue que un contenido que está creado para ser legible por humanos, lo sea también por determinado software.

Aún así, toda la web está sujeta a una licencia Creative Commons Atribución 3.0 que sí permite la reutilización de su contenido siempre que se haga referencia al autor. Por tanto, utilizando mecanismos como el scraping será posible obtener la agenda de conciertos en un formato reutilizable, sin violar ningún tipo de derecho de autor.

## Grüvr

Grüvr es una agenda de conciertos orientada a la geolocalización de éstos sobre un mapa. La página principal detecta nuestra localización geográfica a través de los datos que ofrece nuestro navegador, y muestra un mapa de Google Maps centrado en nuestra localidad, en el que se van mostrando los conciertos más próximos en un radio de 14 kilómetros.

Permite realizar búsquedas de conciertos en base a dos criterios:

- Artista/grupo
- Localización, basada en una localidad y un radio alrededor de ésta.

La geolocalización, además de conferir a Grüvr una gran potencia, es un factor diferenciador frente a otras agendas de conciertos que no ofrecen esta funcionalidad.

Grüvr memoriza los grupos que le interesan al usuario mientras éste consulta sus fechas de concierto, y le permite activar la opción de recibir notificaciones por correo electrónico cuando alguno de estos grupos toque cerca de su localización. También crea un feed personalizado que muestra los próximos conciertos de los artistas favoritos del usuario.

Se puede acceder a los mismos datos que ofrece la web de forma externa, a través de la API, que permite obtener los datos en formato GeoRSS<sup>5</sup>. La forma en que se invocan estos servicios es a través de la URL, existiendo dos sintaxis en función de si la búsqueda es por artista o por localización geográfica, pero siempre siguiendo la arquitectura REST que permite identificar cada recurso de forma unívoca con una URL.

---

<sup>5</sup> GeoRSS es un nuevo estándar para incrustar una localización geográfica dentro de un feed RSS. Dado que un feed RSS es un fichero XML que describen canales de contenido, GeoRSS simplemente añade a RSS un campo más con los datos de longitud y latitud, que le confieren una referencia geográfica.

## 2.e. Tecnología

---

### 2.e.i. Twitter

Twitter es una herramienta social de micro-blogging, que permite al usuario enviar mensajes basados en texto de un máximo de 140 caracteres en tiempo real.

Las actualizaciones se muestran en la página del perfil de usuario, de forma que son accesibles para cualquier usuario que esté conectado a Twitter en ese momento. El emisor del mensaje puede decidir si dicho mensaje estará accesible a cualquiera o sólo a los usuarios de su red social de amigos.

El usuario puede recibir actualizaciones a través de la web de Twitter, mensajería instantánea, SMS, RSS, correo electrónico, o a través de aplicaciones.

Dada la sencillez del servicio, Twitter ha dejado de ser sólo un mecanismo personal para comunicarse con la red social de amigos, para convertirse en una herramienta de comunicación instantánea con el resto del mundo. Es utilizado por gran cantidad de empresas para ofrecer comunicados en tiempo real, noticias, e incluso publicidad.

### 2.e.ii. Yahoo! Pipes

Yahoo! Pipes es una aplicación web para la creación de mashups. A través de una interfaz gráfica de usuario permite construir aplicaciones que integran feeds<sup>6</sup>, páginas web y otros servicios.

Mediante la interfaz gráfica se extrae la información de las fuentes primarias, y uniendo varias herramientas mediante “pipes” o tuberías permite adecuar dicha información y darle forma para al fin obtener el formato de salida deseado.

Algunos ejemplos de “pipes” sencillos que se pueden generar son los siguientes:

- Obtener las noticias del día de los principales diarios nacionales, ordenarlas por fecha, y restringir los resultados a las 50 noticias más recientes.
- Obtener un listado de noticias de la sección internacional y geolocalizarlas en un mapa.

---

<sup>6</sup> **Feed** se refiere a un sistema de sindicación web, que permite difundir noticias o anotaciones, conteniendo un título y un resumen del contenido, mediante ficheros en formatos específicos que permiten la reutilización de la información.

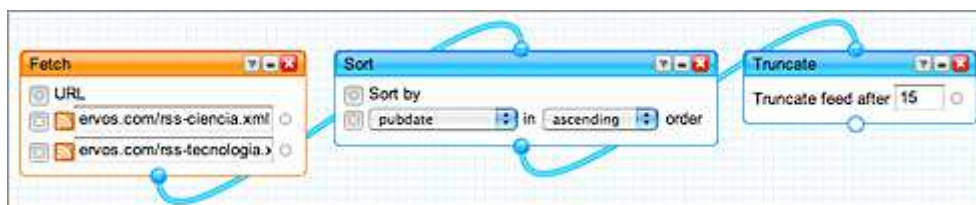
Además, el servicio permite utilizar unas “pipes” dentro de otras, lo que le confiere una gran potencia.

Las “pipes” que crea el usuario pueden ser etiquetadas y compartidas con el resto de los usuarios o, por el contrario, usarlas de forma privada.

Una vez creado, el “pipe” se almacena en la propia plataforma de Yahoo! Pipes y por tanto, es un servicio ejecutado en el servidor. Esta característica tiene ventajas pero también inconvenientes:

- Al estar almacenado en el servidor, permite su acceso desde cualquier terminal, sin necesidad de tener que transportar el programa, y ejecutarlo en cada máquina en la que se desee usar.
- Esto también hace que la confiabilidad sea un concepto que depende de la estabilidad del servidor, de modo que si éste no está accesible, el usuario no podrá acceder a su aplicación hasta que el servidor se restablezca.
- Al ser ejecutado desde la web, hace que sea un servicio multiplataforma, que no depende de un sistema operativo ni de un framework para funcionar. Únicamente es necesario un navegador web con acceso a Internet, y que soporte AJAX si queremos editar el “pipe”.

Actualmente, Yahoo! Pipes es un servicio web en fase beta, de forma que no es una aplicación completamente terminada, y por tanto, puede tener errores inesperados, cortes en el servicio, etc.; pero eso no impide obtener toda la funcionalidad de las herramientas existentes.



**Ilustración 2** Captura de la interfaz de desarrollo

## API

Yahoo! Pipes permite la integración de las “pipes” en webs, blogs, etc, pero además, sigue la arquitectura REST, que permite pasar los parámetros de entrada a la aplicación mediante la URL, y ofrece como formatos de salida RSS, JSON y KML.

### *2.e.iii. Songbird*

Songbird es una aplicación que combina las funciones de reproductor multimedia y navegador web. Es de uso libre y multiplataforma (disponible para Windows, Mac y Linux), y se encuentra en fase beta desde que se comenzó a desarrollar en el año 2006. Está desarrollado por el grupo “Pioneros de lo inevitable”.



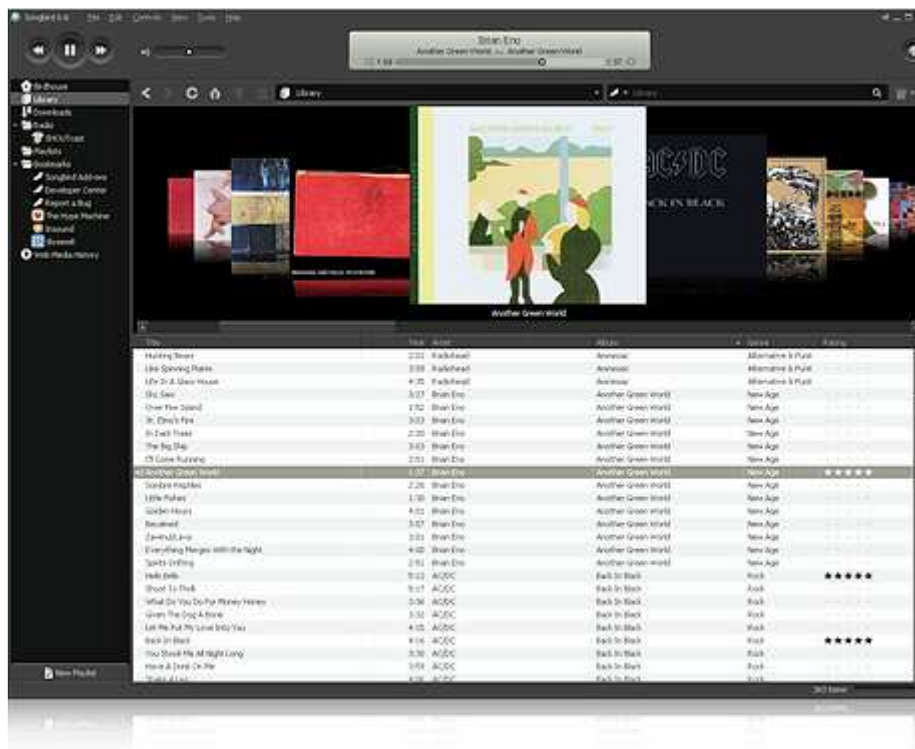


Ilustración 3. Captura de Songbird

Utiliza la plataforma XULRunner de Mozilla para el navegador web, y el cliente VideoLAN para la reproducción de ficheros multimedia. Además utiliza una versión de SQLite para la gestión de la biblioteca multimedia.

Las principales características de la aplicación son:

- En webs con contenido multimedia crea **listas de reproducción de forma dinámica**, cuyo contenido puede ser reproducido o descargado. Es útil para mp3blogs, tiendas de música, sistemas de recomendación musical, buscadores de música, etc.
- Compatibilidad **multiplataforma** con varios sistemas operativos: Microsoft Windows, Mac OS y Linux.
- Capacidad de reproducir multitud de **formatos de audio**.
- Capacidad de **personalización del aspecto** del reproductor mediante el uso de plumas (feathers), creadas utilizando el lenguaje CSS<sup>7</sup>.
- Dado que utiliza el motor XULRunner, hereda la arquitectura de las **extensiones** de Mozilla, útiles para combinar la funcionalidad de reproductor multimedia con la de navegador web.
- **Biblioteca multimedia** que es capaz de indexar y gestionar toda la música que el usuario tenga en su PC.

<sup>7</sup> CSS Cascade Style Sheets, en español Hojas de Estilo en Cascada, es un lenguaje utilizado para modelar el estilo de un contenido determinado.

- Capacidad de **sincronización con reproductores externos** de música, como iPod.
- Capacidad de crear **listas de reproducción** personalizadas.
- Capacidad de convertirse en **mini-reproductor**.
- **Actualizaciones** automáticas.
- **Atajos de teclado**.
- **Suscripciones RSS**<sup>8</sup>.
- Gestión de **favoritos** del usuario.

## Extensiones

Como hemos comentado, Songbird permite la creación de **extensiones** personalizadas, ya que se basa en el motor XULRunner de Mozilla. Esto también posibilita el integrar algunas de las extensiones para otros navegadores basados en la misma arquitectura en Songbird.

Las extensiones se distribuyen en el formato **XPI**, que realmente es un fichero comprimido en ZIP con la extensión cambiada. Se utiliza el lenguaje XUL para definir las interfaces de usuario, JavaScript para la programación y CSS para definir el aspecto de las extensiones.

Para el desarrollo de la interfaz de usuario se utiliza el lenguaje **XUL**. XUL es el acrónimo de XML-based User-interface Lenguaje o Lenguaje basado en XML para la interfaz de usuario. Básicamente se trata de un lenguaje de marcado, basado en XML, mediante el cuál se pueden definir interfaces de usuario.

Al contrario que XML, XUL no es un estándar, y tiene un ámbito de aplicación **limitado** y algo **excluyente**, dado que fue concebido para definir las interfaces de usuario para el software de Mozilla y por tanto, sólo es válido para software derivado de arquitecturas de Mozilla, como es el caso de Songbird. Su principal ventaja es que sirve para simplificar la definición de los interfaces de usuario. Además es multiplataforma.

Para la programación de las extensiones se utiliza **JavaScript**. Es un lenguaje poco potente, pero bastante sencillo, y complementado con los objetos XPCOM (Cross-platform Component Object Model) adquiere esa potencia que le falta.

Mediante **CSS** se define el estilo de las extensiones.

---

<sup>8</sup> RSS Really Simple Syndication, una forma de feed.

## 2.f. Extracción de contenido estático (scraping)

---

Existen ocasiones en las que se presenta la necesidad de reutilizar el contenido de una determinada web que, aunque sus derechos permiten esta reutilización, no ofrece los mecanismos adecuados, es decir, el contenido no tiene la condición de reutilizable. Por ese motivo se crearon los “web scrapers”, aplicaciones destinadas a extraer el contenido de páginas estáticas y convertirlo a otro formato válido para ser reutilizado.

Un ejemplo típico de scraping es una web en formato HTML, que mediante un scraper, se convierte a formato RSS para poder ser utilizado por un lector de feeds.

En nuestro caso analizaremos dos aplicaciones de scraping: Dapper y OpenKapow.

### 2.f.i. Dapper

Dapper es un scraper totalmente orientado a la web, lo que significa que incluso el proceso de diseño del *scraper* sea realizado desde el navegador. Esto le confiere flexibilidad, ya que el usuario no depende de una aplicación y por tanto lo convierte en multiplataforma. Sin embargo, también tiene defectos, como la limitación en el consumo de recursos, la limitación a la hora de crear “robots” complicados, etc.

La aplicación permite crear “robots” de forma sencilla, siguiendo un asistente de cinco pasos, que consisten en:

- Seleccionar la página de origen
- Seleccionar los parámetros que recibe, en caso de hacerlo.
- Seleccionar el contenido que se extraerá.
- Previsualizar el feed.
- Guardar el feed.

Sólo es necesario crear un usuario en Dapper para poder guardar los feeds creados, existiendo la posibilidad de hacerlos públicos o mantenerlos privados, de forma que sólo esté accesible para el usuario.

Una vez creado el *scraper*, este queda almacenado en los servidores de Dapper, y quedará accesible a través de una URL. Permite además escoger entre varios formatos de salida, como XML, RSS iCal, etc. Estas dos condiciones hacen que Dapper siga la arquitectura REST para los scrapers creados.

## 2.f.ii. OpenKapow RoboMaker

Openkapow es otra herramienta de scraping, o una plataforma abierta de servicio, como ellos mismos le llaman, que permite construir servicios y ejecutarlos desde el sitio web de Openkapow. Los “robots”, como llaman al *scraper*, son capaces de acceder a la web de origen y extraer sus datos, así como interactuar con la interfaz de usuario de dicho sitio web.

La forma de actuar del robot vendrá definida por los valores de entrada (ej. Usuario y contraseña), y en base a esto el robot produce una salida.

La peculiaridad que diferencia Openkapow de Dapper es que el proceso de creación del robot se hace mediante una aplicación llamada RoboMaker, la cual requiere instalación (está disponible para Windows y para Linux). Esto tiene ventajas, pero también inconvenientes:

- Ventajas:
  - Al ser una aplicación instalable, las posibilidades durante el diseño del *scraper* que ofrece son superiores a las que se pueden obtener a través de una aplicación web.
- Inconvenientes:
  - Necesidad de descargar e instalar el programa (unos 150MB).
  - Dependencia de las plataformas para las que está creado el programa. Por ejemplo, no se pueden diseñar *scrapers* con RoboMaker en Mac.

Una vez creado el robot, se puede publicar desde la propia aplicación de forma que el nuevo servicio creado quedará accesible desde una URL, y podrá devolver el resultado en varios formatos de salida. (XML, JSON, XHTML, texto, etc.).

# 3. Aplicación

## 3.a. Introducción

---

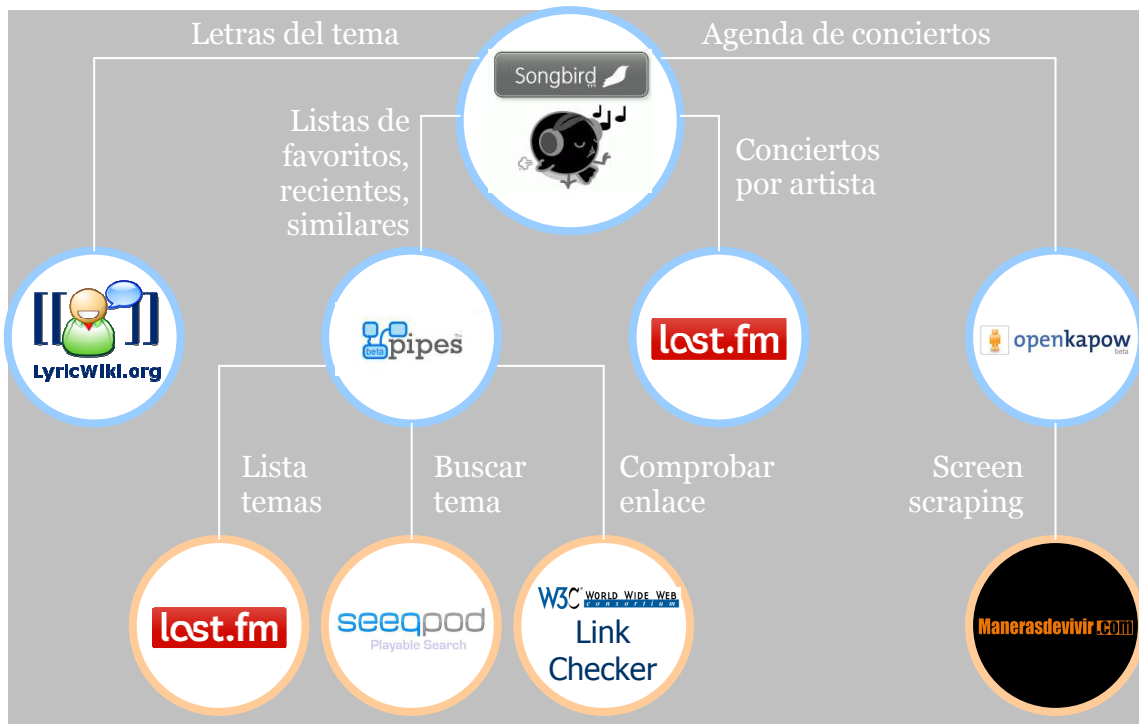
Nuestra aplicación se divide en tres grandes bloques:

- **Descarga de música con Yahoo! Pipes.** Este bloque comprende todos los mashups creados con Yahoo! Pipes que permiten:
  - En base a un usuario de Last.fm, ofrecer descarga de los temas favoritos del usuario, los favoritos de sus amigos, los favoritos de sus vecinos, y los temas que ha escuchado más recientemente.
  - En base al nombre de un artista y un tema, ofrecer un enlace para descargar el tema, y la descarga de una serie de temas que estén relacionados con el primero.
- **Creación de un feed con RoboMaker.** En este apartado utilizaremos la agenda de conciertos de una web de música española para extraer su contenido y convertirlo en contenido reutilizable, de forma que tendremos un listado de conciertos en un formato legible por máquinas.
- **Integración con Songbird.** Crearemos una extensión para la aplicación Songbird, que sea capaz de combinar los componentes anteriores de forma homogénea, añadiendo alguna funcionalidad extra.

## 3.b. Arquitectura

En este esquema se puede observar la comunicación entre los distintos componentes de nuestra aplicación.

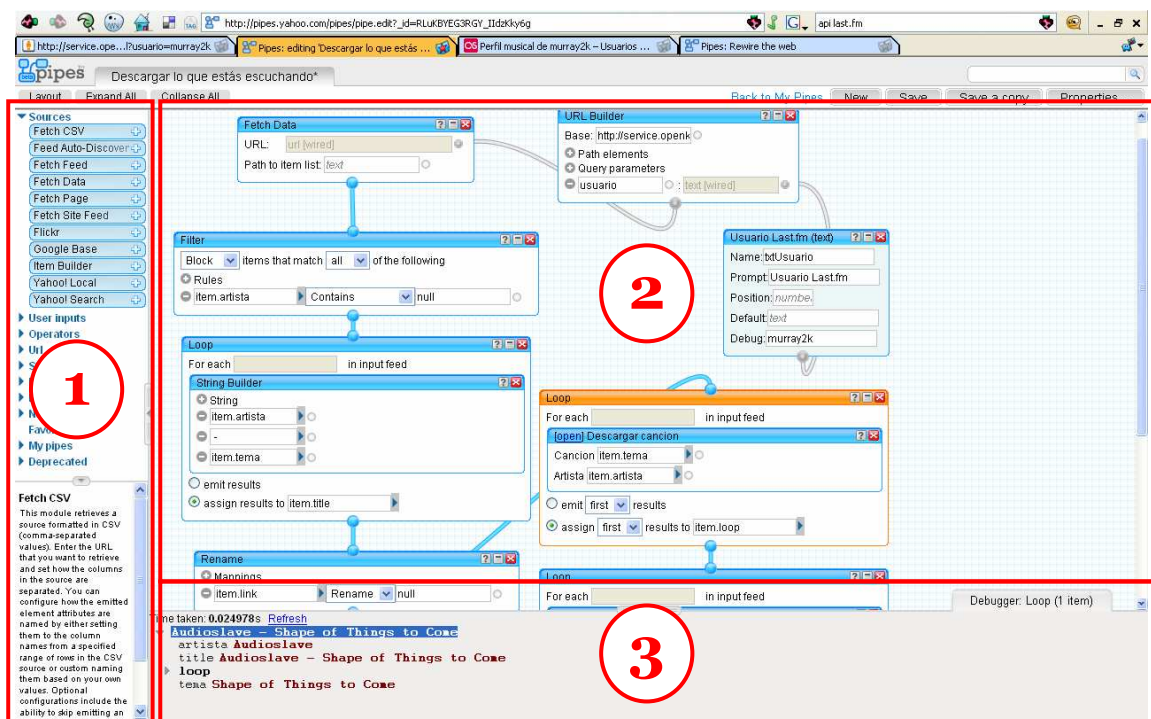
Observamos como el núcleo es Songbird, que se encarga de comunicarse con el resto de componentes.



### 3.c. Descargar canciones con Yahoo! Pipes

Ante el objetivo de realizar búsquedas de temas musicales, comprobar que los resultados de las búsquedas contienen enlaces válidos, y devolver un enlace para la descarga de cada tema; así como repetir este proceso de forma masiva para listas de temas, se ha optado por el uso de Yahoo! Pipes, dada la sencillez con la que permite crear este tipo de aplicaciones, sin necesidad de programar ni de tener excesivos conocimientos técnicos, ahorrando así gran cantidad de tiempo.

#### 3.c.i. Entorno de desarrollo



El entorno de desarrollo es muy sencillo de utilizar. Está dividido en tres partes:

1. **Barra de herramientas.** En ella se sitúan todas las herramientas disponibles para crear el pipe. Están divididas en:
  - a. Sources. Contiene herramientas para la extracción de datos de formatos de sindicación, páginas estáticas, etc.
  - b. User inputs. Herramientas para entrada de datos de usuario.
  - c. Operators. Aquí se encuentran todas las herramientas que nos permitirán operar con el feed que vamos a crear. Contiene herramientas de todo tipo, como filtro, contador, bucle, expresiones regulares, o incluso una herramienta para llamar a un servicio web externo.



- d. URL. Únicamente contiene la herramienta “URL Builder”, útil para la generación de URLs de servicios REST.
  - e. String. Herramientas para trabajar con cadenas de caracteres. Permite cortar, unir cadenas, extraer subcadenas, trabajar con expresiones regulares, traducir. Es importante puntualizar que la entrada de este tipo de herramientas no puede ser un feed, sino una cadena simple. Si queremos usarla para trabajar con cadenas en el campo de un feed determinado, debemos incluirla dentro de la herramienta “loop”.
  - f. Date. Herramientas para crear o dar formato a campos de fecha.
  - g. Location. Contiene la herramienta “Location Builder”, muy útil para la regularización de localizaciones. A partir de unos datos de entrada, como una dirección, lo transforma en una localización regularizada, incluyendo código postal, longitud, latitud, etc.
  - h. Number. Contiene la herramienta “Simple Math” que permite hacer cálculos matemáticos básicos.
  - i. Favourites. Contiene las herramientas que el usuario haya incluido dentro de sus favoritas.
  - j. My pipes. Contiene los pipes que el usuario haya creado, de forma que pueden ser reutilizadas dentro de otros pipes.
  - k. Deprecated. Contiene pipes que con el tiempo han caído en desuso por haber sido sustituidas por otras nuevas más eficaces.
2. **Diseño.** En esta zona se encuentra el diseño del pipe. Está compuesto por herramientas, unidas entre sí mediante tuberías o enlaces que interconectan las entradas y salidas de las herramientas, y siempre ha de acabar en un “pipe output”.
  3. **Debug.** Es una parte muy importante del entorno de desarrollo, ya que permite ver en todo momento la salida que genera la herramienta que tenemos seleccionada en ese momento. Pulsando en la opción “Refresh” podemos ver la salida más actualizada.

### 3.c.ii. Requisitos

Las necesidades que se nos presentan son las siguientes:

- Ofrecer un listado de los temas favoritos de un usuario determinado de Last.fm, así como un enlace para la descarga de éste, en caso de estar disponible.
- Ofrecer un listado de temas recientemente escuchados por un usuario determinado de Last.fm, así como un enlace para la descarga de éste, en caso de estar disponible.
- Ofrecer un listado de los temas favoritos de los amigos en la red social de un determinado usuario de Last.fm, así como un enlace para la descarga de éste, en caso de estar disponible.

- Ofrecer un listado de los temas favoritos de los vecinos en la red social de un determinado usuario de Last.fm, así como un enlace para la descarga de éste, en caso de estar disponible.
- Ofrecer un listado de los temas relacionados musicalmente con un tema determinado, así como un enlace para la descarga de éste, en caso de estar disponible.

Para ello, previamente se presenta la necesidad de construir un par de herramientas que permitan:

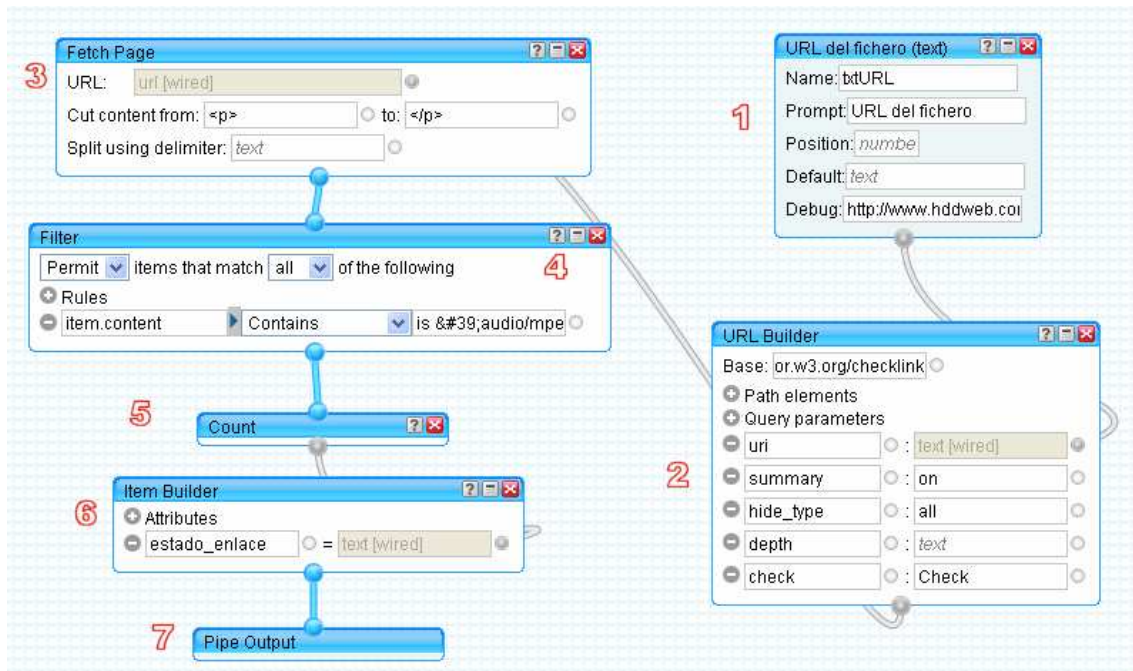
1. Comprobar que la URL que enlace con la descarga de un fichero esté disponible.
2. Realizar el proceso de:
  - En base a un artista y un tema recibidos como parámetro, realizar una búsqueda contra la API de un buscador web de temas musicales en formato mp3.
  - Comprobar qué resultados de la búsqueda son válidos para su descarga mediante la herramienta comentada anteriormente.
  - Devolver un enlace para la descarga del tema.

### *3.c.iii. Solución*

#### Comprobar la disponibilidad de un enlace

[http://pipes.yahoo.com/pipes/pipe.info?\\_id=faa09a950054501e3e22401663c42e70](http://pipes.yahoo.com/pipes/pipe.info?_id=faa09a950054501e3e22401663c42e70)

Ante la necesidad de comprobar el estado de un enlace a un fichero mp3, se pensó en una solución basada en la combinación de Yahoo! Pipes con la herramienta Link Checker, una aplicación web creada por el World Wide Web Consortium, que permite, en base a una URL pasada como parámetro, comprobar que el recurso al que apunta dicha URL está disponible, e incluso que se trata de un fichero de audio/video, una ventaja en nuestro caso.



Nuestra aplicación de Yahoo! Pipes hace lo siguiente:

1. Recibe como parámetro la URL del fichero mp3, por ejemplo “http://www.hddweb.com/90737/Avril\_Lavigne\_-\_Why.mp3”.
2. En base a esta URL, construye una nueva URL para realizar la consulta a Link Checker.
3. Realiza la consulta a Link Checker.
4. Filtra el resultado devuelto por Link Checker, en busca del texto “&#39;audio/mpeg&#39;”, que indicaría que la URL apunta a un fichero válido de audio.
5. Cuenta los resultados. Aquí solo puede darse el caso de que sean 0 o 1, en caso de que haya pasado el filtro del paso 4 o no.
6. Devuelve el resultado de la cuenta, de forma que devolverá 0 si el enlace no es válido, o 1 si el enlace sí es válido.

## Descargar un tema

<http://pipes.yahoo.com/murray2k/descargacancion>

En este caso era necesario que, en base al nombre de un artista y un tema recibidos como parámetro, se devuelva un enlace para la descarga de dicho tema en caso de encontrarlo y que se encuentre disponible.

Para ello, primero era necesario localizar un buscador web de MP3 que ofreciera una API pública para poder hacer uso de su motor de búsqueda, o que al menos ofreciera la posibilidad legal de reutilizar los resultados que ofrezca su buscador en formato HTML. Los candidatos eran varios, descritos en el apartado 2.d.i.:

- **Skreemr**

<http://www.skreemr.com/>

Skreemr es un potente buscador de mp3, con las ventajas de que los resultados son bastante fiables. Utiliza una característica propia muy interesante llamada Audiorank, que ordena los temas en base a diversos criterios de calidad. El problema es que no ofrece ningún tipo de API pública, ni la posibilidad de utilizar los resultados que ofrece su buscador web.

- **Woonz**

<http://www.woonz.com/>

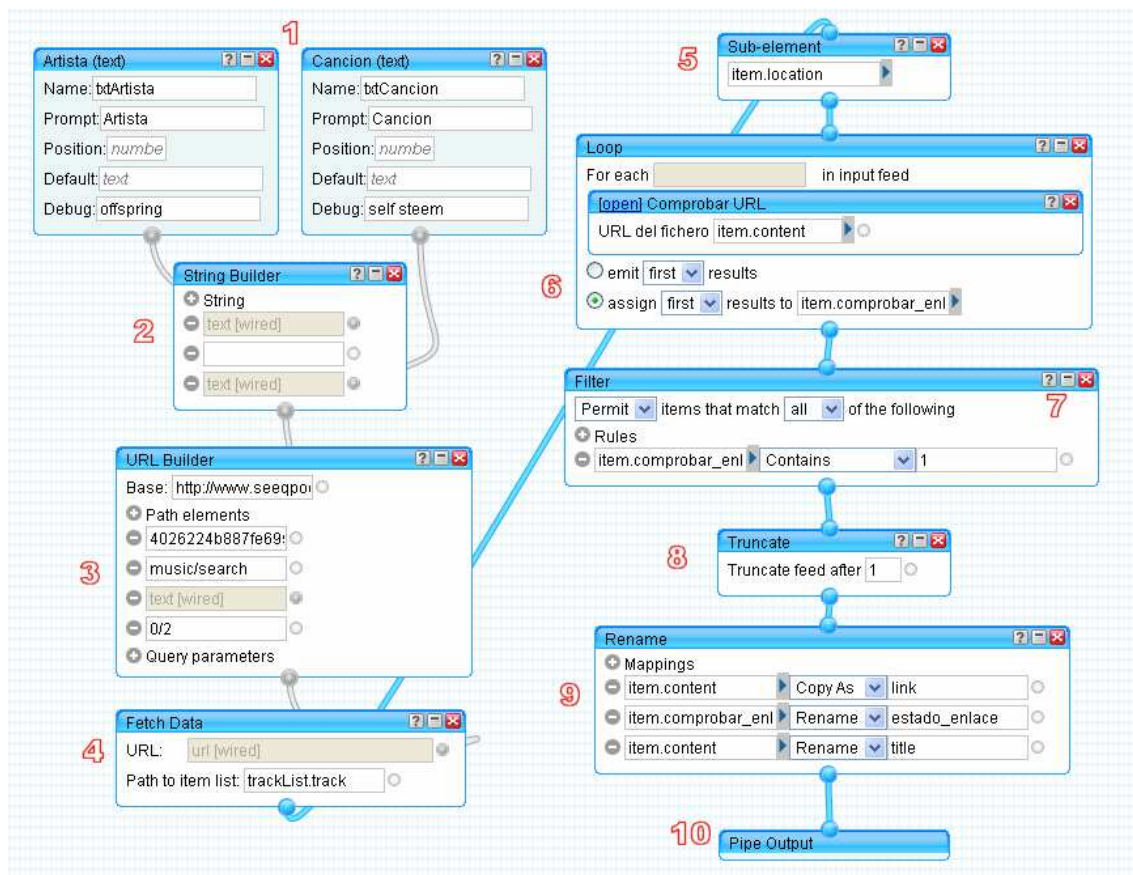
Woonz es otro buscador web de mp3 que ofrece gran cantidad de resultados, aunque no del todo fiables. Se desechó por no ofrecer API.

- **Seeqpod**

<http://www.seeqpod.com/>

Seeqpod es otro potente buscador de mp3, que además de ofrecer gran cantidad de resultados, relativamente precisos, ofrece una API pública que sólo requiere tener un usuario registrado, algo que se hace rápidamente y de forma gratuita.

Finalmente se optó por Seeqpod, dado que era el único que ofrecía una API pública y los resultados eran relativamente precisos.



Una vez seleccionado el buscador, era necesario que desde Yahoo! Pipes se realizara todo el proceso, desde recibir los parámetros de búsqueda hasta devolver un enlace comprobado:

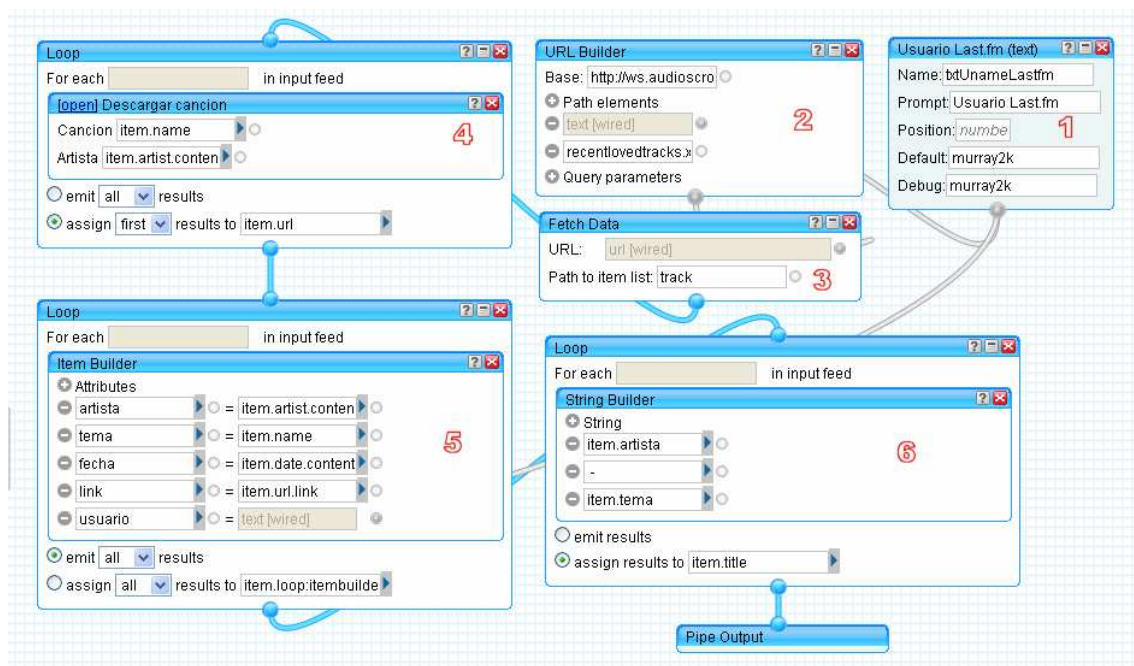
1. Yahoo! Pipes recibe como parámetro el nombre del artista y del tema.
2. Construye una cadena única en base a las dos cadenas anteriores.
3. Construye la URL para la consulta a la API de Seeqpod, solicitando como máximo 2 resultados.
4. Realiza la petición a la API de Seeqpod.
5. De los resultados, se extrae únicamente la URL del fichero mp3.
6. Para cada resultado (puede haber 0, 1 ó 2 resultados), se llama al Pipe “Comprobar URL” comentado en el apartado 3.b.i., que se encarga de verificar que la URL apunta a un fichero válido de audio.
7. Filtra los resultados, de forma que sólo los que contengan un enlace válido pasan.
8. Trunca los resultados para que haya un máximo de un resultado. Es decir, cero resultados en caso de no haber encontrado ningún enlace válido, o un resultado en caso de encontrar un enlace válido.
9. Renombra algunos campos del feed de respuesta.

## Descargar favoritos de Last.fm

<http://pipes.yahoo.com/murray2k/bajarfavoritoslastfm>

El objetivo en este caso es conectarse con la API de Last.fm, solicitar un listado de los temas favoritos del usuario que se recibe como parámetro, buscar los temas uno por uno en Seeqpod y añadir la URL en aquellos para los que se encuentre un enlace para descarga.

Para ello será necesario utilizar la API de Last.fm, de uso libre, y con una interfaz muy sencilla, que identifica los recursos mediante la URL y ofrece varios formatos de respuesta, aunque nosotros utilizaremos XML.



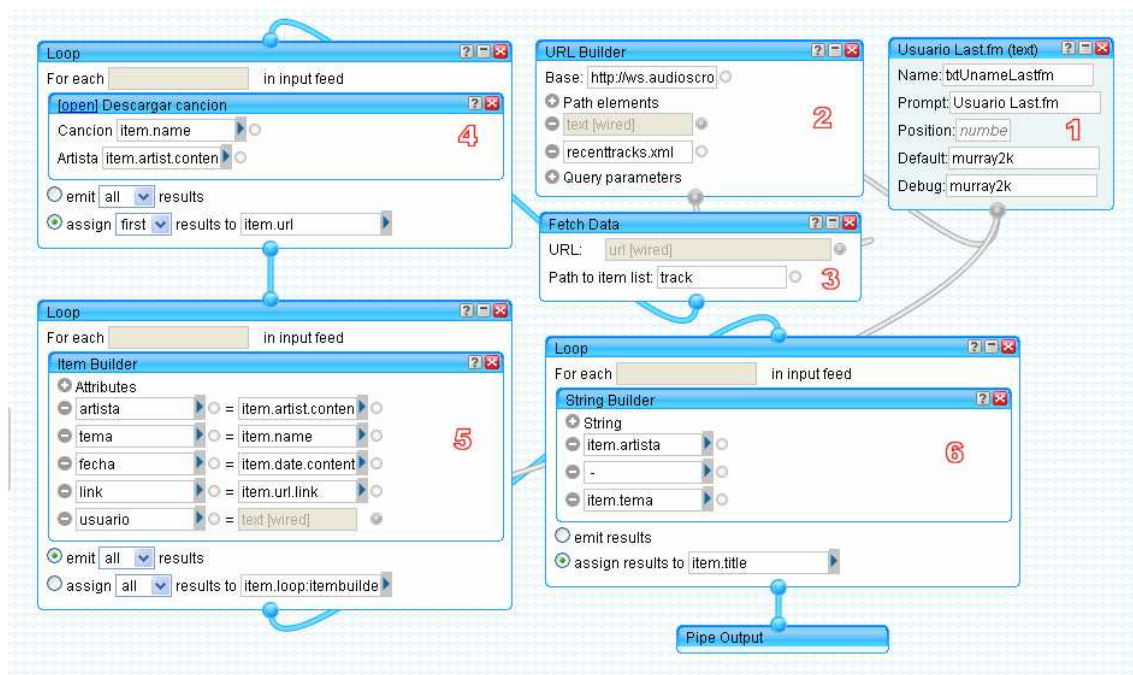
El proceso que realiza nuestra aplicación de Yahoo! Pipes es el siguiente:

1. Recibe como parámetro un usuario de Last.fm.
2. Construye la URL para la petición a la API de Last.fm de los temas favoritos del usuario.
3. Realiza la petición.
4. Para cada uno de los temas devueltos en el resultado, llama al pipe “Descargar canción”, que añadirá a cada tema un enlace para la descarga en caso de encontrarlo.
5. Realiza algunos cambios en los nombres de los campos y su contenido.
6. Pone como título de cada tema el artista y tema.

## Descargar temas recientes de Last.fm

<http://pipes.yahoo.com/murray2k/recientes>

En este caso el proceso es muy parecido a la descarga de favoritos de Last.fm. La diferencia es que en la petición a Yahoo! Pipes se solicita una lista de temas recientes del usuario en lugar de un listado de temas favoritos.

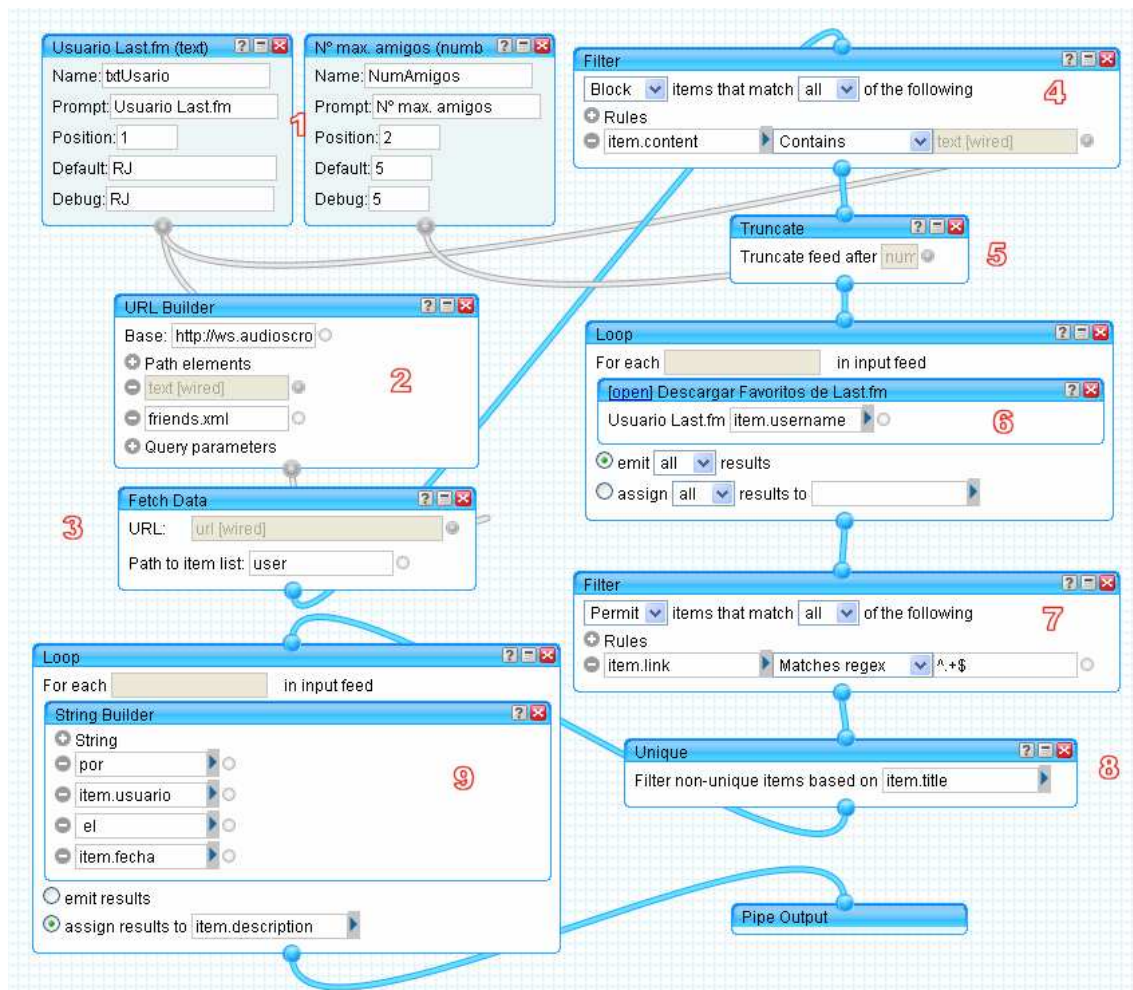


## Descargar favoritos de amigos de Last.fm

[http://pipes.yahoo.com/murray2k/fav\\_amigos](http://pipes.yahoo.com/murray2k/fav_amigos)

En este caso, nuestro pipe debe solicitar a Last.fm un listado de amigos de un usuario recibido como parámetro y reutilizar el pipe “Descargar favoritos de Last.fm” para descargar los favoritos de cada uno de ellos.



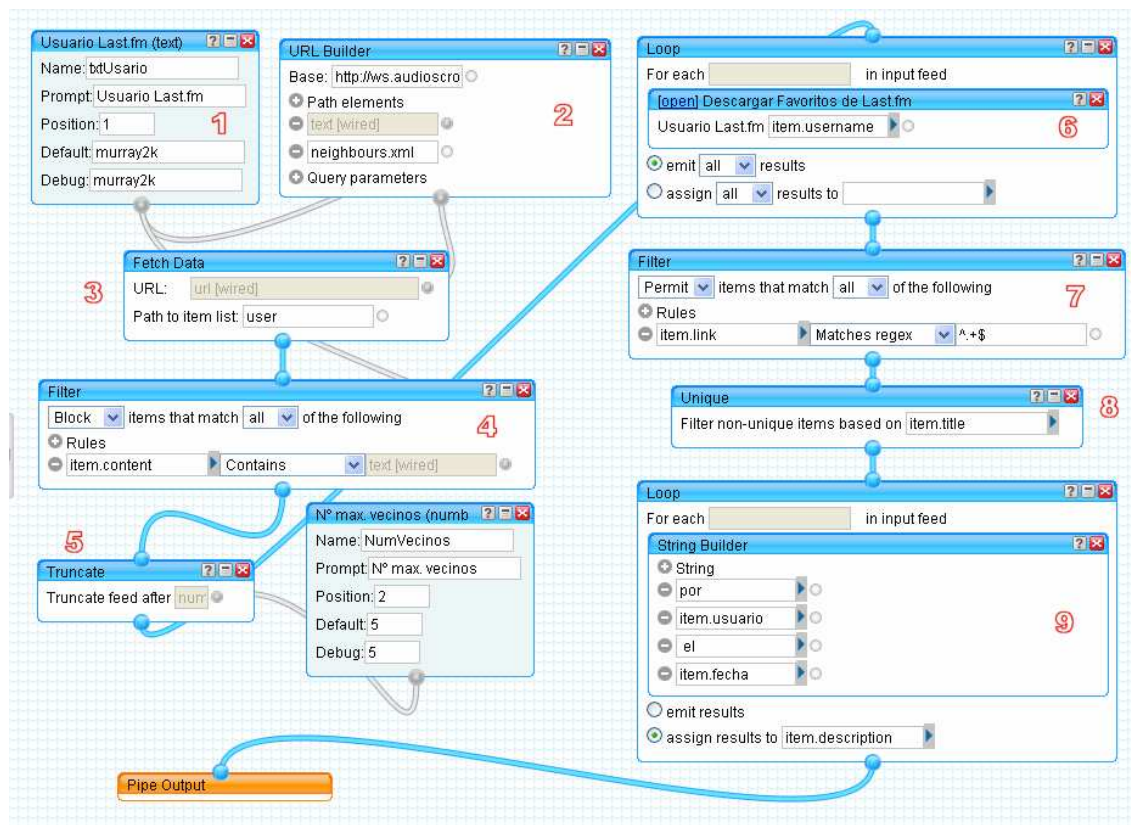


1. Recibe como parámetro un usuario de Last.fm y un número máximo de amigos.
2. Construye la URL para la petición a la API de Last.fm de un listado de amigos del usuario.
3. Realiza la petición.
4. Elimina al propio usuario de la lista de amigos.
5. Trunca los amigos al número que se haya establecido en el segundo parámetro.
6. Para cada uno de los amigos, llama al pipe “Descargar favoritos de Last.fm”, y añade los resultados a cada amigo.
7. Elimina los resultados que no contengan un enlace para evitar una gran cantidad de resultados y obtener así un listado de temas con enlace.
8. Realiza un filtro para que no haya temas repetidos.
9. añade la fecha y el usuario que ha escuchado ese tema antes de marcarlo como favorito.

## Descargar favoritos de vecinos de Last.fm

[http://pipes.yahoo.com/murray2k/fav\\_vecinos](http://pipes.yahoo.com/murray2k/fav_vecinos)

En este caso el objetivo es muy parecido al del pipe “Descargar favoritos de amigos de Last.fm”, con la diferencia de que solicitamos a la API de Last.fm un listado de vecinos del usuario en lugar de un listado de amigos.

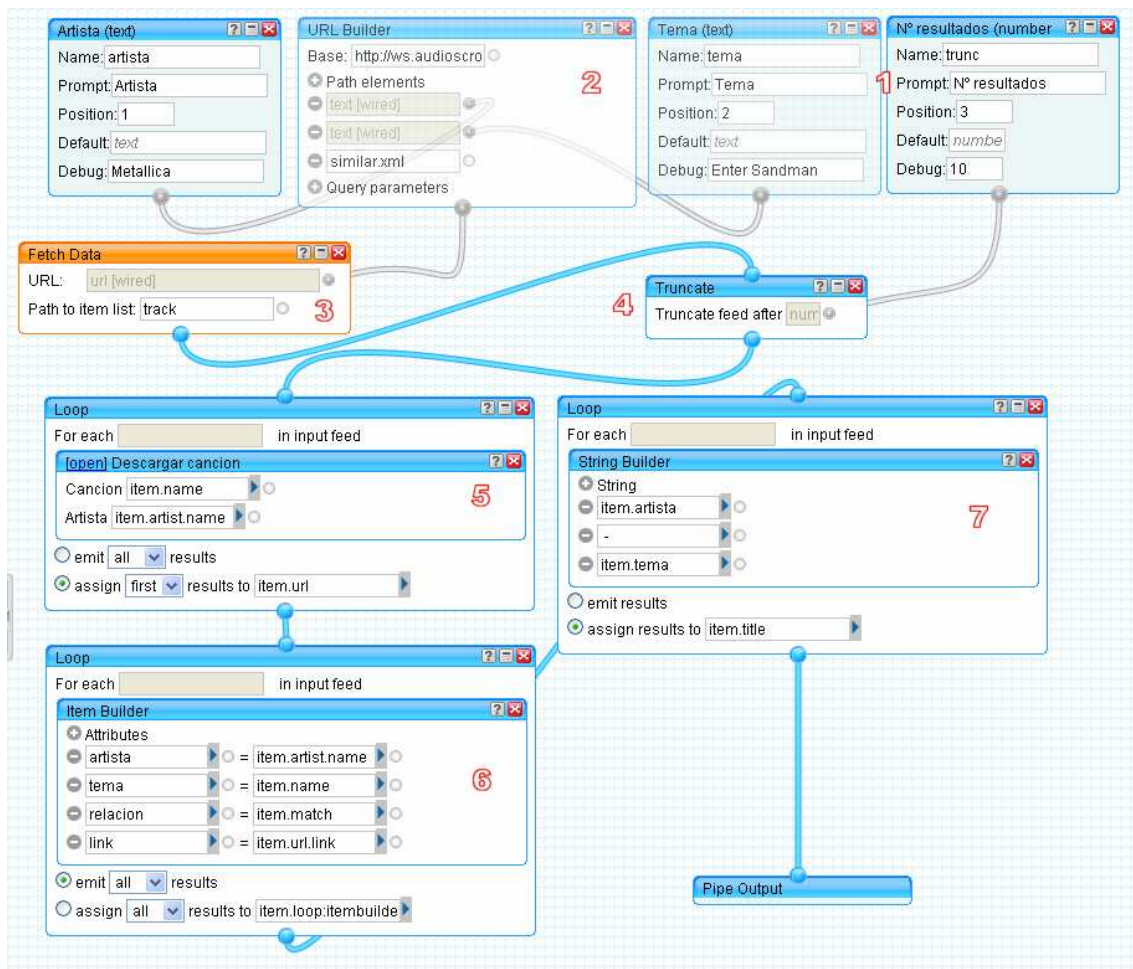


## Descargar temas similares

[http://pipes.yahoo.com/murray2k/temas\\_similares](http://pipes.yahoo.com/murray2k/temas_similares)

El pipe “Descargar temas similares” debe recibir como parámetro el nombre de un artista y un tema y, en base a ello, solicitar a la API de Last.fm un listado de temas similares. Una vez recibido ese listado será necesario llamar para cada tema al pipe “Descargar tema” que se encargará de proporcionar un enlace para descarga en caso de encontrarlo.





1. Recibe como parámetro el nombre de un artista, el nombre de un tema y un número máximo de resultados.
2. Construye la URL para la consulta a la API de Last.fm.
3. Realiza la consulta.
4. Trunca los resultados a un máximo de temas determinados por el parámetro recibido en el paso 1.
5. Para cada uno de los temas llama al pipe “Descargar canción”, que añadirá un enlace al tema en caso de encontrarlo.
6. Realiza cambios en algunos campos.
7. Construye el título de cada entrada del feed en base a artista y tema.

## Invocación de los ‘pipes’ contruidos

Yahoo! Pipes sigue la arquitectura REST para sus pipes. De esta forma, podemos invocar cada una de las pipes creadas mediante una URL, del tipo:

[http://pipes.yahoo.com/pipes/pipe.run?\\_id=460551735ebce046121d65a1cf7b7cac&render=json&txtUnameLastfm=murray2k](http://pipes.yahoo.com/pipes/pipe.run?_id=460551735ebce046121d65a1cf7b7cac&render=json&txtUnameLastfm=murray2k)

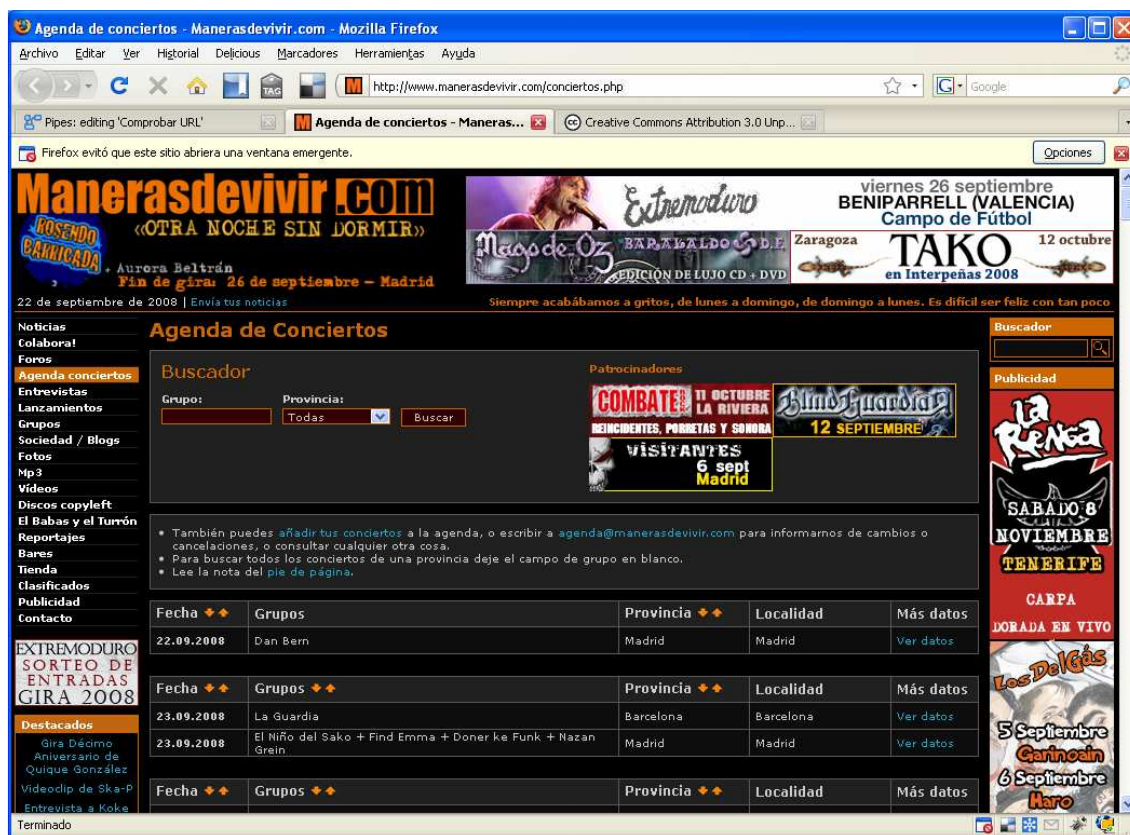
En la cual invocamos el servicio de Yahoo! Pipes que se encarga de invocar las pipes (<http://pipes.yahoo.com/pipes/pipe.run>) y le pasamos ciertos parámetros:

1. `id=460551735ebce046121d65a1cf7b7cac`. El identificador de nuestro pipe.
2. `render=json`. Indica el formato de salida, en nuestro caso JSON.
3. Resto de parámetros que utilice nuestro pipe. En este caso un usuario de Last.fm (`txtUsernameLastfm=murray2k`)

## 3.d. Creando un feed con RoboMaker

### 3.d.i. Requisitos

El objetivo en este caso es poder extraer el listado de conciertos que ofrece la web Manerasdevivir (<http://www.manerasdevivir.com>) para poder reutilizarlo en nuestra aplicación. El problema que se presenta es que, aunque dicho listado se ofrece en la web con una licencia que permite su reutilización, la web no ofrece los mecanismos para hacerlo, dado que presenta dicha información en una página estática HTML.



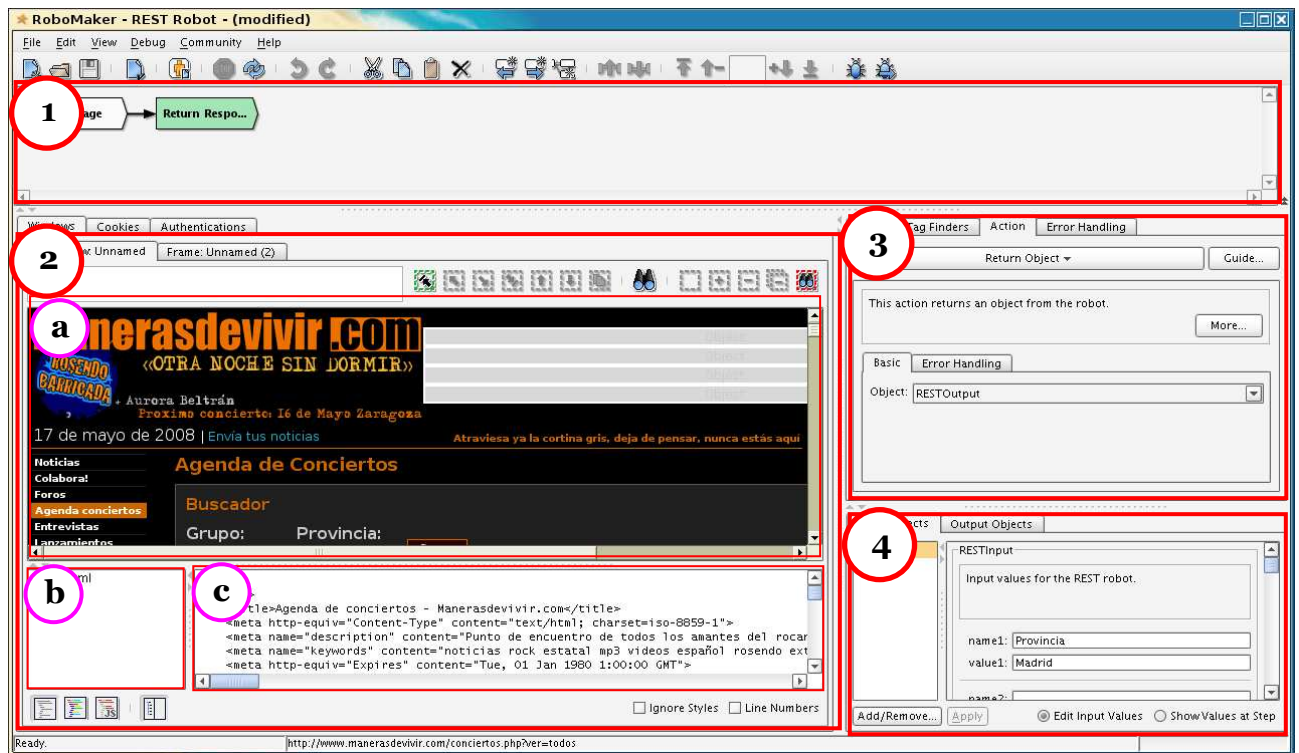
Para poder reutilizar dicha información tendremos que hacer uso de una herramienta de scraping, capaz de analizar el contenido estático de una página y devolver como salida el mismo contenido en formato dinámico.

El programa elegido en nuestro caso es RoboMaker de la compañía OpenKapow (<http://openkapow.com>), por su potencia y por ofrecer varios formatos de salida, además de contar con un servidor donde alojar nuestro scraper para poder acceder a él de forma remota desde cualquier máquina.

Para poder crear nuestro scraper o robot será necesario descargar RoboMaker, disponible para las plataformas Windows y Linux. En nuestro caso optamos por la plataforma Linux dado que en Windows se produjeron problemas que impedían la ejecución del programa.

### 3.d.ii. Entorno de desarrollo

La interfaz principal del programa es la siguiente:



En la que podemos observar:

- 1) Flujo. Muestra el flujo de ejecución del robot.
- 2) Ventana web, que a su vez se divide en:
  - a) Navegador.
  - b) Vista en árbol.
  - c) Código.
- 3) Pasos. Donde se pueden modificar las propiedades de cada paso del robot.
- 4) Objetos. Donde se pueden ver/modificar los objetos de entrada/salida.

En la ventana web (2) vemos tanto el contenido como el código de la web sobre la que queremos hacer scraping.

En el flujo (1) podemos observar todos los pasos que contiene nuestro robot para el análisis y extracción de datos de la web.

En los pasos (3) modificamos de forma individual cada uno de los pasos que se observan en el flujo.

### 3.d.iii. Solución

Mediante esta herramienta construimos nuestro robot, que se encargará de conectarse a la web de conciertos de Manerasdevivir:

<http://www.manerasdevivir.com/conciertos.php>

E ir recorriendo su estructura de tablas extrayendo los datos de cada uno de los conciertos, mediante un bucle que analiza cada una de las filas de la tabla, y estructura dichos datos para darles un formato reutilizable.

Opcionalmente, nuestro robot puede extraer el listado completo de conciertos, o hacerlo para una provincia concreta que reciba como parámetro.

El formato para invocar nuestro robot sigue la arquitectura REST. La sintaxis es la siguiente:

**`http://service.openkapow.com/murray2k/conciertos.<Formato_salida>?Provincia=<Provincia>`**

- **Formato\_salida** puede adoptar los siguientes valores: xml, html, csv, json o xhtml.
- **Provincia** puede tomar el valor de cualquier provincia española, así como Ceuta y Melilla siguiendo la codificación ASCII<sup>9</sup>.

---

<sup>9</sup> ASCII: Acrónimo de American Standard Code for Information Interchange. Código de caracteres basado en el alfabetolatino.

## 3.e. Integración con Songbird

---

### 3.e.i. Requisitos

Una vez tenemos creados los principales componentes de nuestra aplicación, se presenta la necesidad de integrarlos todos de forma homogénea. Contamos con los siguientes componentes:

- Descargar un tema.
- Descargar temas favoritos de un usuario de Last.fm.
- Descargar temas recientes de un usuario de Last.fm.
- Descargar temas favoritos de los amigos de un usuario de Last.fm.
- Descargar temas favoritos de los vecinos de un usuario de Last.fm.
- Descargar temas similares a un tema dado.
- Agenda de conciertos en español.

Observando los componentes podemos darnos cuenta de que uno de los puntos fuertes de la aplicación es ofrecer grandes cantidades de información en base a un usuario de Last.fm.

Sin embargo, quedan algunos componentes como descargar un tema, descargar temas similares a uno dado o la agenda de conciertos. Es necesario integrarlos en nuestra aplicación, y el nexo de unión puede ser orientarlos a la música que el usuario escucha en un momento dado, y ofrecer descarga de temas similares, del propio tema, y un listado de conciertos del grupo que suena.

Para ello, utilizamos Songbird. Como se ha comentado anteriormente, Songbird es una aplicación que une la funcionalidad de un reproductor de música con la de un navegador web; es de código abierto y ofrece la posibilidad de crear y utilizar extensiones que añaden funcionalidad al programa.

Por ello decidimos crear una extensión que será el corazón de nuestra aplicación; el nexo de unión para los componentes creados. Esto implica que, mientras todos los componentes creados hasta el momento son ejecutados de forma remota, el núcleo de nuestra aplicación se ejecutará en cliente.

Las ventajas de nuestra solución son:

- Mucha mayor potencia a la hora de poder realizar la integración.
- La posibilidad de ofrecer información actualizada relacionada con la música que el usuario escucha en su PC.

Mientras que los inconvenientes son:

- Es necesario instalar Songbird. Esto le hace depender de las plataformas para las que Songbird ha sido creado (Windows, Linux y Mac).
- Es necesario instalar la extensión para Songbird.

Para el desarrollo de nuestra extensión utilizaremos los lenguajes:

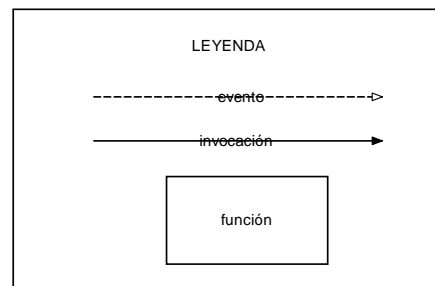
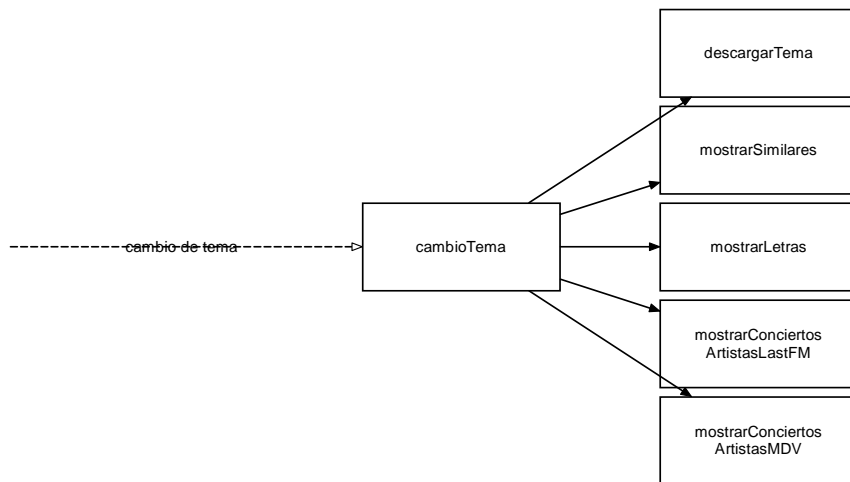
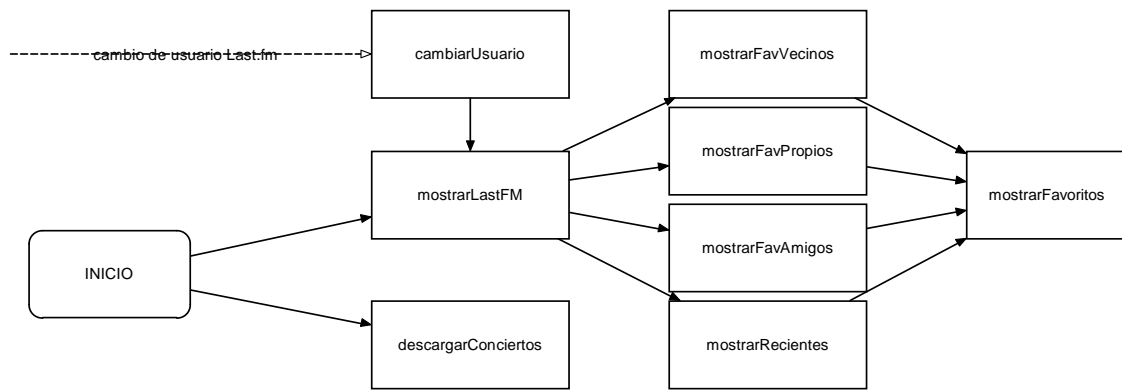
- **XUL** (XML-based User-interface Lenguaje), un lenguaje creado por Mozilla para la creación de interfaces de forma sencilla usando el lenguaje de marcado XML.
- **Javascript**. Para la programación de nuestra extensión. Es un lenguaje limitado en potencia pero relativamente sencillo de utilizar.
- **CSS** para la definición de estilos sobre el aspecto de nuestra extensión.

Finalmente, una vez creada la extensión, es necesario comprimirla en un fichero ZIP y renombrarla al formato XPI para que adquiriera la condición de extensión instalable en Songbird.

### *3.e.ii. Solución*

A continuación explicaremos la estructura básica del código Javascript de nuestra extensión. Dicha extensión se divide en dos grandes bloques: uno relativo al usuario Last.fm, y otro relativo al tema que el usuario está escuchando.

En el siguiente esquema se puede observar la estructura del código:



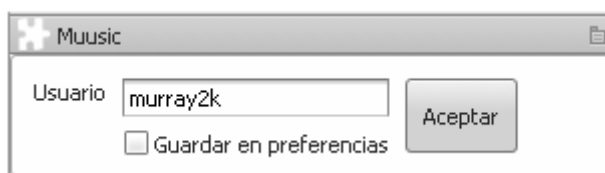


Observamos como al inicializar la extensión llamamos a las funciones `mostrarLastFM` y `descargarConciertos`:

- **descargarConciertos** hace la petición a OpenKapow del listado de conciertos de Manerasdevivir. Se hace una sola vez al iniciar la aplicación dado el largo tiempo de proceso que requiere para el robot y el tamaño de los datos a recibir, unos 80 KB. Después guarda los datos recibidos en una variable global para que puedan ser consultados a posteriori cada vez que se cambie de tema y ofrecer así los conciertos del artista en caso de encontrarse.
- **mostrarLastFM** se encarga de extraer el usuario de Last.fm de las preferencias de la extensión, y llamar a las cuatro funciones encargadas de mostrar los listados de temas relacionados con ese usuario:
  - **mostrarFavVecinos** pide a Yahoo! Pipes el listado de los temas favoritos disponibles para descarga de los vecinos del usuario Last.fm, y llama a la función `mostrarFavoritos` para que los muestre en la pestaña “Vecinos” de nuestra extensión.
  - **mostrarFavPropios** pide a Yahoo! Pipes el listado de los temas favoritos de usuario Last.fm, y llama a la función `mostrarFavoritos` para que los muestre en la pestaña “Favoritos” de nuestra extensión.
  - **mostrarFavAmigos** pide a Yahoo! Pipes el listado de los temas favoritos disponibles para descarga de los amigos del usuario Last.fm, y llama a la función `mostrarFavoritos` para que los muestre en la pestaña de “Amigos” de nuestra extensión.
  - **mostrarRecientes** pide a Yahoo! Pipes el listado de los diez temas recientemente escuchados por el usuario Last.fm, y llama a la función `mostrarFavoritos` para que los muestre en la pestaña de “Recientes de nuestra extensión.
  - **mostrarFavoritos** es la función encargada de mostrar los listados de temas que recibe como parámetro en una tabla, y poner un enlace para descargar el tema en caso de existir.



Es posible cambiar de usuario Last.fm una vez que la extensión ya ha arrancado. Para ello se han habilitado unos controles en la interfaz que permiten introducir un nuevo usuario y elegir si queremos que se guarde en preferencias.



Al presionar el botón Aceptar enviamos un evento a nuestra función `cambiarUsuario`:

- **`cambiarUsuario`** extrae el usuario del cuadro de texto y lo guarda en preferencias en caso de haber sido seleccionada esta opción. Después llama a la función `mostrarLastFM`, que se encarga de mostrar los listados de temas relativos al nuevo usuario.

Por otro lado, cada vez que cambia el tema en reproducción, la función `cambioTema` recibe un evento:

- **`cambioTema`** extrae los datos del tema en reproducción de Songbird, y los muestra en la pestaña “Escuchando”. Se conecta con la API de Last.fm y busca la portada del álbum. Si la encuentra la muestra. Después llama a las funciones:
  - **`descargaTema`** recibe el nombre de un artista y un tema como parámetro, se conecta con Yahoo! Pipes para pedir un enlace de descarga. En caso de recibirlo, lo muestra en la pestaña “Escuchando”. Caso contrario, muestra un mensaje diciendo que no se encuentra un enlace para la descarga del tema.
  - **`mostrarSimilares`** recibe el nombre de un artista y un tema como parámetro, y se conecta con Yahoo! Pipes para pedir un listado de temas relacionados. Después muestra en la pestaña “Similares” el listado de temas recibido.
  - **`mostrarLetras`** recibe el nombre de un artista y un tema como parámetro, y se conecta con la API de LyricWiki.org para solicitarle las letras del tema. En caso de recibirlas, las muestra en la pestaña “Letras”, caso contrario, muestra un mensaje indicando que no ha sido posible localizar las letras del tema.
  - **`mostrarConciertosArtistaLastFM`** recibe el nombre de un artista como parámetro, y solicita la API de Last.fm un listado de conciertos de ese artista. Si recibe resultados, los muestra en la pestaña “Conciertos”, caso contrario muestra un mensaje indicando que no ha sido posible encontrar conciertos de ese artista en Last.fm.
  - **`mostrarConciertosArtistaMDV`** recibe el nombre de un artista como parámetro, y consulta en el listado de conciertos descargado anteriormente y almacenado en una variable global en busca del artista. Si encuentra conciertos, los muestra en la pestaña “Conciertos”, caso contrario, muestra un mensaje diciendo que no encuentra conciertos en Manerasdevivir.

## 4. Discusión

## 4.a. Introducción

---

A continuación verteremos nuestras impresiones acerca de las aplicaciones que hemos utilizado para el desarrollo de nuestra aplicación, ventajas e inconvenientes, problemas con los que nos hemos topado durante los periodos de investigación y desarrollo, e impresiones generales de cada una de las aplicaciones.

## 4.b. Yahoo! Pipes

---

El servicio web de Yahoo! Pipes fue lanzado en febrero de 2007 y actualmente continúa en fase beta, lo que significa que puede contener fallos y errores, dado que no es una versión definitiva. Aquí se describen algunos fallos que se han encontrado durante su uso:

1. En el entorno de desarrollo, hay ocasiones en que la pantalla de debug no se actualiza aún pulsando la opción de “Refresh”, y puede ser necesario pulsarlo más de una vez.
2. Durante el desarrollo, de forma arbitraria puede ocurrir un error que hace que en el debug sólo pueda verse el mensaje “Preview failed”. Cuando esto pasa, es imposible continuar con el desarrollo, y es necesario volver a una versión anterior guardada del pipe para poder seguir trabajando, ya que tampoco permite guardar el pipe en el estado actual.
3. Lentitud. Tanto durante el desarrollo, como durante la ejecución, se observa bastante lentitud, más cuando se trata de un pipe que debe realizar varias peticiones web. Esto es inevitable, dado que se trata de un servicio web, y no depende de la aplicación, sino de los tiempos de respuesta de servicios de terceros, proveedores de Internet, etc.
4. Baja confiabilidad. Cuando el número de peticiones web que el pipe realiza es alto, además de lentitud, se puede obtener un mensaje de error como “Error de servidor”, sin obtener la respuesta esperada.
5. Baja tasa de actualización. Se observa cómo, al utilizar un pipe para obtener datos que cambian con frecuencia, no se obtiene la respuesta deseada, pudiendo obtener datos antiguos que no se corresponden. Esto posiblemente pasa porque el servicio guarda en caché la información consultada anteriormente, y al obtener una consulta igual a la hecha hace unos instantes, extrae la respuesta de su caché en lugar de volver a consultar la información. Esto impide que pueda usarse para servicios de actualización frecuente.
6. Al tratarse de un servicio ofrecido por Yahoo! a través de Internet, nuestro pipe depende de que dicho servicio se siga ofreciendo de la misma manera que se ha hecho hasta el momento. Si la compañía cierra, cambia de política, o es absorbida por otra compañía con otros intereses, es posible que nuestro pipe no pueda seguir funcionando por retirada del servicio, cambio en las condiciones, etc.
7. Escasez de formatos de salida. Yahoo! Pipes ofrece formatos de salida específicos para ciertos servicios, como Bloglines, Rojo, Netvibes, Newsgator o MyAOL. También ofrece salida en formato RSS (lo que limita los campos de salida a las especificaciones de dicho formato), JSON y PHP. Se echan de menos formatos de salida como XML, CSV, o microformatos para ciertos tipos de información.

Yahoo! Pipes es una herramienta que se diferencia de cualquier otra, con muchas posibilidades a cambio de poco esfuerzo, y nada de programación. Algunos de los problemas que se han encontrado son inherentes a un servicio ofrecido a través de Internet, y otros son fallos que quedarán solucionados cuando la aplicación termine su fase beta. Por tanto se puede decir que en términos generales las impresiones con Yahoo! Pipes son satisfactorias.

## 4.c. Screen scraping con RoboMaker

---

Nuestro objetivo en este caso era hacer screen scraping sobre la agenda de conciertos de una web de música. La aplicación elegida para ello es RoboMaker de la compañía OpenKapow.

En el campo del screen scraping no existen muchas alternativas conocidas, excepto Dapper, que tiene ciertas limitaciones al tratarse de una aplicación web. Sin embargo, buscando información relacionada con la creación de mashups, tuve la suerte de dar con un blog<sup>10</sup> en el que se explicaba la creación de un mashup que extrae los datos de embalses españoles de la web del Ministerio de Medio Ambiente utilizando un scraper llamado Open Kapow Robomaker. Tras descargar dicho programa, instalarlo y probarlo, decidí utilizarlo para la extracción de datos de agenda de conciertos.

El uso del programa es bastante peculiar, y **requiere un periodo de aprendizaje** y estudio de otros robots antes de poder empezar a crear uno propio. Sin embargo, una vez adquiridos unos conocimientos básicos, ofrece grandes posibilidades.

No obstante, se hecha en **falta algo de documentación** relacionada con el uso del programa. La ayuda que incorpora es bastante escueta y en pocas ocasiones cumple su función, ya que se limita a describir la funcionalidad de las cosas, y no a explicar su modo de uso. No existe ningún manual que explique el funcionamiento de la aplicación, y lo único que se puede encontrar en la web de Openkapow son algunos tutoriales con ejemplos para la creación de robots sencillos, con los cuales aprender a utilizar el programa de forma básica.

Aún así, se puede decir que hoy por hoy RoboMaker es el **programa más potente que hemos encontrado en su campo**, ya que permite construir ‘scrapers’ sin necesidad de conocimientos de programación, lo cual dice mucho a su favor. La única alternativa que permitiría haber hecho la misma tarea hubiera sido programar nuestro propio ‘scraper’, una tarea que posiblemente hubiera requerido mucho más esfuerzo que el que ha conllevado la creación del ‘scraper’ con RoboMaker.

---

<sup>10</sup> Artículo “Cómo hacer un mashup y no morir en el intento”, del blog Franchu’s Lair <http://franchu.net/2007/06/02/como-hacer-un-mashup-y-no-morir-en-el-intento/>

## 4.d. Songbird

---

La idea de partida con Songbird era investigar las posibilidades que ofrecía a la hora de integrar una extensión que permitiera acceder a la información del reproductor, además de comunicarse con nuestros mashup creados con Yahoo! Pipes y RoboMaker, y fuera capaz de ofrecer la información que nos interesa.

Para ello, investigué la forma en que se crean las extensiones en Songbird. Tras leer algo de documentación en la web de documentación del programa<sup>11</sup>, descubrí que la creación de una extensión no presenta gran dificultad. Para crear la interfaz basta con aprender a usar un sencillo lenguaje de definición de interfaces creado por Mozilla, XUL. Para la parte de programación, se utiliza JavaScript, un lenguaje sencillo para crear aplicaciones no excesivamente complicadas.

Tras instalar algunas extensiones descargadas de la web del programa e investigar su código, me decidí a crear mi propia extensión. Para ello, Songbird incorpora un pequeño asistente que crea la estructura de ficheros y directorios necesaria para tener una extensión ‘modelo’. Una vez creada, sólo hay que añadir, quitar o modificar lo que creamos conveniente.

Tras haber terminado la extensión, se puede decir que Songbird **ofrece una potente plataforma para la creación de extensiones**, que además es sencilla. La única dificultad que podemos encontrar es que, debido al corto periodo de vida de la aplicación -todavía se encuentra en fase beta-, existe poca documentación. Sin embargo, dado que la web de documentación es un Wiki es previsible que poco a poco vaya creciendo su contenido.

---

<sup>11</sup> Web de documentación para el desarrollador de Songbird:  
[http://wiki.songbirdnest.com/Developer/Developer\\_Intro](http://wiki.songbirdnest.com/Developer/Developer_Intro)



# 5. Conclusiones

## 5.a. Introducción

---

Llegados a este punto, podemos analizar la consecución de objetivos:

- Hemos analizado diversos **servicios orientados a la música** que se ofrecen a través de Internet, así como las posibilidades que ofrecen para reutilizar la información ofertada a través de APIs.
- Hemos analizado los servicios que ofrece **Last.fm** a través de su API, y los hemos explotado de forma diversa en nuestra aplicación:
  - Extrayendo datos del perfil de usuario: temas favoritos, temas recientes, listado de amigos, listado de vecinos.
  - Extrayendo datos del álbum.
  - Extrayendo datos geográficos de conciertos.
- Hemos analizado varias **herramientas de creación de mashups**, como Yahoo! Pipes, Open Kapow RoboMaker, Dapper, comentando sus características principales, ventajas y desventajas. Hemos analizado más en profundidad dos de ellas: Yahoo! Pipes y RoboMaker.
- Hemos analizado el funcionamiento de la herramienta de creación de mashups **Yahoo! Pipes**. De hecho, gran parte de nuestra aplicación se basa en esta herramienta, y en la explotación que hace de los recursos que ofrecen Last.fm y otros servicios. Finalmente hemos extraído algunas impresiones obtenidas durante el uso de dicha herramienta.
- Hemos investigado las distintas posibilidades que teníamos de extraer el contenido estático de la agenda de conciertos Manerasdevivir, mediante la técnica de **screen scraping**. Finalmente, hemos escogido una herramienta llamada **RoboMaker** para esta labor, con la cual hemos obtenido resultados satisfactorios.
- Hemos integrado todos los componentes de nuestra aplicación en una extensión para **Songbird**, la cual integra la posibilidad de conocer la música en reproducción con la creación de extensiones que permiten la comunicación con distintos servicios web. Esto nos ha permitido integrar de forma idónea todos los componentes de nuestra aplicación, de una forma homogénea.

De forma global podemos decir que aún se podría mejorar para alcanzar una mayor facilidad a la hora de crear aplicaciones heterogéneas. Para ello se proponen una serie de medidas que serían recomendables para cualquier Web que ofrezca información con posibilidad de ser reutilizada:

- **Separar la el contenido del diseño.** O lo que es lo mismo, separar la información del código. Esto permitiría reutilizar la información sin necesidad de recurrir a técnicas como el screen scraping, que consumen tiempo de programación, recursos, y no siempre ofrecen la precisión necesaria.

- **Añadir contenido semántico a la información.** Una vez alcanzado el hito de separar la información del diseño, se puede ir más allá, añadiendo metadatos que doten a la información de un significado. Dotando a la información de estructura, relaciones, etc. se puede conseguir que la información sea evaluable por máquinas sin necesidad de la interacción humana.
- **Utilizar estándares.** El uso de estándares es importante a la hora de reutilizar la información.

Durante el desarrollo de nuestra aplicación han ido surgiendo nuevas ideas interesantes, que podrían haber sido incorporadas a la aplicación. Sin embargo, debido a los problemas de tiempo y a que no habían sido incluidas en el plan inicial, no han podido ser incorporadas. No obstante, en el próximo apartado comentaremos dichas ideas, quedando reflejadas como líneas futuras de desarrollo.

## 5.b. Líneas futuras de desarrollo

---

En este apartado hablaremos de posibles mejoras sobre nuestra aplicación. Mejoras que no han sido incorporadas debido a que nuestra aplicación se hubiera complicado en exceso, y debido a las limitaciones de tiempo, no ha sido posible incorporar.

### 5.b.i. *Desambiguación*

Uno de los problemas que se nos plantea a la hora de identificar un tema es que tenemos que fiarnos de la información que el reproductor nos proporciona. El reproductor, a su vez, se fía de la información que incorpora el propio fichero de audio, de forma que si no es correcta, o no está bien escrita, nos será mal informada.

Por este motivo podemos obtener:

- Nombres de temas que no se corresponden con el tema que está sonando.
- Nombres de temas del estilo ‘track01’ por no haber sido correctamente rellenada la información del fichero de audio.
- Nombres de temas mal escritos.

Esto provoca que nuestra aplicación no funcione correctamente, ya que se basa en la información que recibe del reproductor, y si ésta no es correcta, no será capaz de mostrar la información relativa al tema.

Para solventar este problema no existe una solución perfecta, al menos por el momento. Sin embargo, existen algunos métodos que tienen una cierta precisión. Uno de ellos es una herramienta en forma de pequeña aplicación de línea de comandos que proporciona Last.fm, y que es capaz de, en base a un fichero de música, crear una huella de identidad y consultar al sistema para obtener la información del tema.

Podríamos por tanto integrar esta aplicación en nuestra extensión, de forma que cada vez que se empezara a escuchar un tema nuevo, se pasara la ruta del fichero a esta aplicación, y ésta nos devolviera los datos correctos del tema. De esta forma, suponiendo que los datos del tema que se nos proporcionen sean siempre los correctos, podríamos decir que habríamos salvado el escoyo de no conocer el nombre del tema, tenerlo mal escrito, etc.

### 5.b.ii. *Conciertos por localización*

Nuestra aplicación ofrece actualmente un listado de conciertos en relación al artista que está sonando. Sin embargo, el feed que hemos creado con una agenda de conciertos a

nivel nacional, unido a la agenda de conciertos del sistema Last.fm, nos proporciona una completa agenda de conciertos de artistas nacionales e internacionales.

Una de las futuras líneas de desarrollo podría ser extraer la localización geográfica del sistema, y ofrecer un listado de **conciertos cercanos a la localización del usuario**.

Para ello, deberíamos ser capaces de extrapolar el dato de la localización geográfica a:

- Nuestra agenda de conciertos de Manerasdevivir. Se podrían ofrecer los conciertos que coincidieran en localidad, y aquellos que fueran en la misma provincia.
- Agenda de conciertos de Last.fm. A través de su API ofrece dos formas de pasar la localización: por el dato localización, o por latitud/longitud. Lo más interesante sería pasar la localización, y utilizar otro parámetro, distancia, en la que pasaríamos un radio para localizar conciertos dentro de ese radio.

### *5.b.iii. Detección de errores en peticiones HTTP*

Actualmente la aplicación no es capaz de diferenciar los distintos tipos de errores que se pueden producir al hacer peticiones HTTP a los distintos servicios que hemos creado (Yahoo! Pipes, OpenKapow, etc). Simplemente los gestiona todos de la misma forma, independientemente de que sea un error de tiempo excedido, un error de servidor, etc.

Otra futura línea de desarrollo sería diferenciar entre los distintos tipos de errores que se pueden producir, de forma que se pudiera mostrar información más precisa al usuario. De momento, ante cualquier error se muestra el mensaje “No se encontraron temas” independientemente del error que se produzca.

Para ello habría que diferenciar entre los distintos códigos de error que podemos obtener:

- 1xx Conexión rechazada
- 2xx Operación exitosa
- 3xx Redirección hacia otro URL
- 4xx Error por parte del cliente
- 5xx Error por parte del servidor

Y mostrar un mensaje informativo al usuario en función del error.

#### *5.b.iv. Redifusión de temas en Twitter*

Aprovechando la funcionalidad que ofrece Twitter de sindicación en tiempo real, se puede añadir a la aplicación la funcionalidad de publicar en Twitter la canción que estamos escuchando en ese momento, siempre que el usuario así lo decida mediante la pulsación de un botón. Se puede complementar con la funcionalidad de añadir un comentario personalizado, con las impresiones del usuario sobre el tema en cuestión.

Otra opción es la publicación de forma automática de cada uno de los temas que escucha el usuario, pero quizá esta opción sería demasiado cargante dada la gran cantidad de mensajes que se publicarían en Twitter.

## 6. Bibliografía

- 1 George Coulouris, Jean Dollimore, Tim Kindberg  
**Distributed Systems: Concepts and design (Fourth edition)**  
Addison Wesley
- 2 Sanjiva Weerawarana, Francisco Cubera, Frank Leymann, Tony Storey, Donald F. Ferguson  
**Web Services Platform Architecture**  
Prentice Hall
- 3 **All Music:** sitio oficial  
<http://www.allmusic.com/>  
Medio musical especializado  
Última visita: julio de 2008
- 4 **All Music:** información en Wikipedia  
[http://es.wikipedia.org/wiki/All\\_Music\\_Guide](http://es.wikipedia.org/wiki/All_Music_Guide)  
<http://en.wikipedia.org/wiki/Allmusic>  
Última visita: julio de 2008
- 5 **Foafing the music:** sitio oficial  
<http://foafing-the-music.iua.upf.edu/>  
Sistema de recomendación musical  
Última visita: julio de 2008
- 6 **GeoRSS:** información  
<http://en.wikipedia.org/wiki/GeoRSS>  
<http://georss.org/>  
Última visita: julio de 2008
- 7 **Google Trends**  
<http://www.google.com/trends/>  
Tendencias musicales de los usuarios de Google  
Última visita: julio de 2008
- 8 **Grooveshark:** sitio oficial  
<http://www.grooveshark.com>  
Sistema de recomendación musical  
Última visita: julio de 2008
- 9 **Grooveshark:** artículo  
<http://www.genbeta.com/2007/09/09-grooveshark-recompensa-a-los-usuarios-por-compartir-sus-temas-musicales>  
Artículo extraído del blog Genbeta  
Última visita: julio de 2008
- 10 **Gruvr:** sitio oficial  
<http://gruvr.com/>  
Agenda de conciertos  
Última visita: julio de 2008



11	<b>Last.fm:</b> sitio oficial <a href="http://www.last.fm/">http://www.last.fm/</a> Sistema de recomendación musical Última visita: julio de 2008
12	<b>Last.fm:</b> descripción del API <a href="http://www.lastfm.es/api/">http://www.lastfm.es/api/</a> API del sitio Last.fm Última visita: julio de 2008
13	<b>LyricWiki:</b> sitio oficial <a href="http://www.lyricwiki.org/">http://www.lyricwiki.org/</a> Letras de canciones Última visita: julio de 2008
14	<b>Manerasdevivir:</b> sitio oficial <a href="http://www.manerasdevivir.com/">http://www.manerasdevivir.com/</a> Portal sobre música en español Última visita: julio de 2008
15	<b>Mashup:</b> artículo de IBM <a href="http://www.ibm.com/developerworks/xml/library/x-mashups.html">http://www.ibm.com/developerworks/xml/library/x-mashups.html</a> Última visita: julio de 2008
16	<b>Mashup:</b> información en Wikipedia <a href="http://es.wikipedia.org/wiki/Mashup_(aplicaci%C3%B3n_web_h%C3%ADbrida)">http://es.wikipedia.org/wiki/Mashup_(aplicaci%C3%B3n_web_h%C3%ADbrida)</a> <a href="http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)">http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid)</a> <a href="http://es.wikipedia.org/wiki/AJAX">http://es.wikipedia.org/wiki/AJAX</a> <a href="http://es.wikipedia.org/wiki/Resource_Description_Framework">http://es.wikipedia.org/wiki/Resource_Description_Framework</a> <a href="http://es.wikipedia.org/wiki/RDF_Schema">http://es.wikipedia.org/wiki/RDF_Schema</a> <a href="http://es.wikipedia.org/wiki/Sindicaci%C3%B3n">http://es.wikipedia.org/wiki/Sindicaci%C3%B3n</a> <a href="http://es.wikipedia.org/wiki/Sindicaci%C3%B3n_web">http://es.wikipedia.org/wiki/Sindicaci%C3%B3n_web</a> <a href="http://en.wikipedia.org/wiki/Web_syndication">http://en.wikipedia.org/wiki/Web_syndication</a> <a href="http://es.wikipedia.org/wiki/Rss">http://es.wikipedia.org/wiki/Rss</a> <a href="http://en.wikipedia.org/wiki/Atom_(standard)">http://en.wikipedia.org/wiki/Atom_(standard)</a> Última visita: julio de 2008
17	<b>Microformatos:</b> información en Wikipedia <a href="http://es.wikipedia.org/wiki/Microformatos">http://es.wikipedia.org/wiki/Microformatos</a> <a href="http://en.wikipedia.org/wiki/Microformats">http://en.wikipedia.org/wiki/Microformats</a> Última visita: julio de 2008
18	<b>Myspace:</b> sitio oficial <a href="http://www.myspace.com/">http://www.myspace.com/</a> Herramienta para la creación de perfiles personales Última visita: julio de 2008
19	<b>MyStrands:</b> sitio oficial <a href="http://www.mystrands.com/">http://www.mystrands.com/</a>

	Sistema de recomendación Última visita: julio de 2008
20	<b>PartyStrands:</b> sitio oficial <a href="http://www.partystrands.com/">http://www.partystrands.com/</a> Aplicación musical de tipo social Última visita: julio de 2008
21	<b>Programmable Web APIs</b> <a href="http://www.programmableweb.com/apis">http://www.programmableweb.com/apis</a> Web sobre APIs de servicios web Última visita: julio de 2008
22	<b>Rolling Stone:</b> sitio oficial <a href="http://www.rollingstone.es/">http://www.rollingstone.es/</a> Revista electrónica sobre música Última visita: julio de 2008
23	<b>Rolling Stone:</b> información en Wikipedia <a href="http://es.wikipedia.org/wiki/Rolling_Stone">http://es.wikipedia.org/wiki/Rolling_Stone</a> <a href="http://en.wikipedia.org/wiki/Rolling_Stone">http://en.wikipedia.org/wiki/Rolling_Stone</a> Última visita: julio de 2008
24	<b>Seeqpod:</b> sitio oficial <a href="http://www.seeqpod.com/">http://www.seeqpod.com/</a> Buscador de música Última visita: julio de 2008
25	<b>Servicios Web:</b> información en Wikipedia <a href="http://es.wikipedia.org/wiki/SOAP">http://es.wikipedia.org/wiki/SOAP</a> <a href="http://en.wikipedia.org/wiki/SOAP">http://en.wikipedia.org/wiki/SOAP</a> <a href="http://en.wikipedia.org/wiki/Web_services_protocol_stack">http://en.wikipedia.org/wiki/Web_services_protocol_stack</a> <a href="http://es.wikipedia.org/wiki/Servicios_Web">http://es.wikipedia.org/wiki/Servicios_Web</a> <a href="http://en.wikipedia.org/wiki/Web_service">http://en.wikipedia.org/wiki/Web_service</a> <a href="http://en.wikipedia.org/wiki/Representational_State_Transfer">http://en.wikipedia.org/wiki/Representational_State_Transfer</a> <a href="http://en.wikipedia.org/wiki/Remote_procedure_call">http://en.wikipedia.org/wiki/Remote_procedure_call</a> <a href="http://en.wikipedia.org/wiki/Remote_procedure_call">http://en.wikipedia.org/wiki/Remote_procedure_call</a> <a href="http://en.wikipedia.org/wiki/Universal_Description_Discovery_and_Integration">http://en.wikipedia.org/wiki/Universal_Description_Discovery_and_Integration</a> Última visita: julio de 2008
26	<b>Skreemr:</b> sitio oficial <a href="http://www.skreemr.com/">http://www.skreemr.com/</a> Buscador de música Última visita: julio de 2008
27	<b>SOAP:</b> Información diversa <a href="http://www.w3schools.com/soap/default.asp">http://www.w3schools.com/soap/default.asp</a> <a href="http://www.desarrolloweb.com/articulos/1557.php">http://www.desarrolloweb.com/articulos/1557.php</a> Última visita: julio de 2008

28	<b>Songbird:</b> sitio oficial <a href="http://getsongbird.com/">http://getsongbird.com/</a> Sitio oficial del reproductor de música Última visita: julio de 2008
29	<b>Strands Blog</b> <a href="http://blog.strands.com/">http://blog.strands.com/</a> Blog oficial del sitio Strands.com Última visita: julio de 2008
30	<b>Twitter:</b> sitio oficial <a href="http://www.twitter.com/">http://www.twitter.com/</a> Sistema de micro-blogging Última visita: julio de 2008
31	<b>Wikipedia:</b> sitio oficial <a href="http://es.wikipedia.org/">http://es.wikipedia.org/</a> Enciclopedia en línea Última visita: julio de 2008
32	<b>Woonz:</b> sitio oficial <a href="http://www.woonz.com/">http://www.woonz.com/</a> Buscador de música Última visita: julio de 2008
33	<b>XUL:</b> Mozilla developer center <a href="http://developer.mozilla.org/es/docs/XUL">http://developer.mozilla.org/es/docs/XUL</a> Documentación sobre XUL Última visita: julio de 2008
34	<b>XUL:</b> artículo “Pa xulo, XUL” <a href="http://geekipollas.com/?p=169">http://geekipollas.com/?p=169</a> Artículo sobre XUL de un blog sobre programación Última visita: julio de 2008
35	<b>Youtube:</b> sitio oficial <a href="http://es.youtube.com/">http://es.youtube.com/</a> Web de vídeos Última visita: julio de 2008
36	<b>Youtube:</b> información en Wikipedia <a href="http://es.wikipedia.org/wiki/YouTube">http://es.wikipedia.org/wiki/YouTube</a> <a href="http://en.wikipedia.org/wiki/YouTube">http://en.wikipedia.org/wiki/YouTube</a> Última visita: julio de 2008

## 7 ■ Anexos

## 7.a. Instalación de la extensión para Songbird

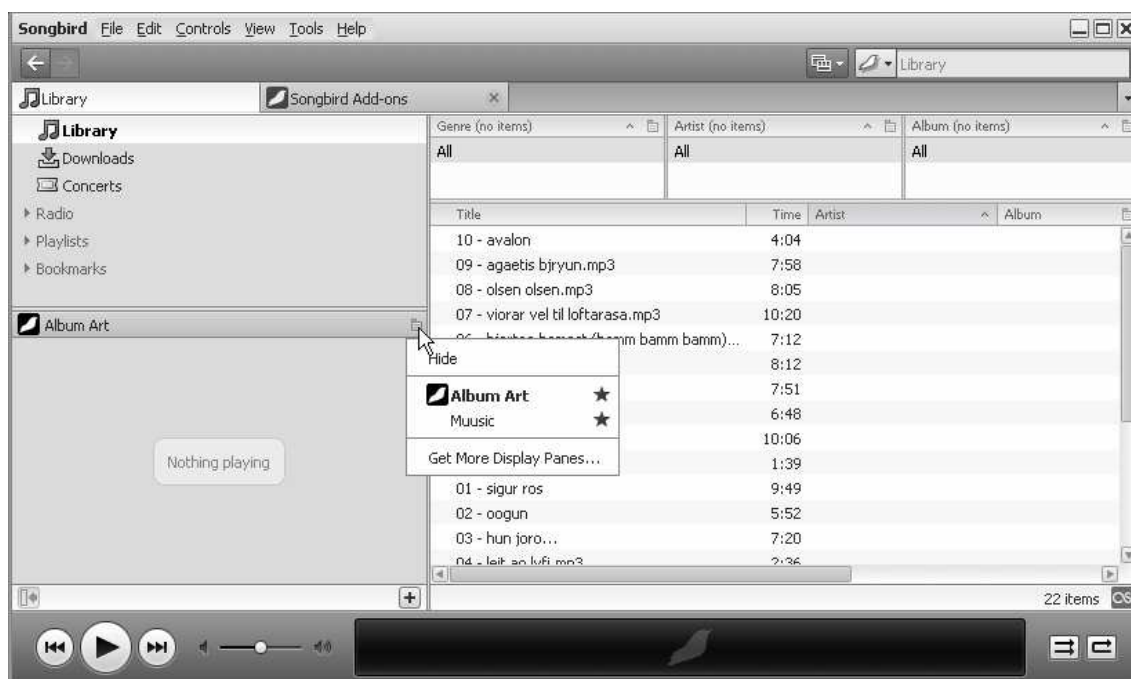
Para poder instalar la extensión, es necesario tener instalada la aplicación Songbird versión 0.7.0 para las plataformas Windows, Linux o Mac. Se adjunta una copia de la aplicación en el CD que acompaña a la documentación. En caso de no disponer del CD, se puede descargar una copia del programa en la siguiente dirección:


<http://www.getsongbird.com>

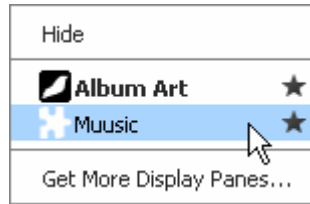
Una vez instalado el programa, será necesario disponer del fichero de instalación para la extensión, “Muusic.xpi”. Se encuentra tanto en el CD que acompaña a la documentación como en la siguiente dirección:

<http://addons.songbirdnest.com/addon/1336>

Para instalar la extensión es necesario ejecutar Songbird. Una vez abierto, debemos dirigirnos al menú “Tools” → “Add-ons”. Se abre una ventana, en la que tendremos que pinchar sobre el botón “Install”. Seleccionamos el fichero “Muusic.xpi”, y tras aceptar una confirmación, la extensión quedará instalada. Tras esto es necesario reiniciar la aplicación Songbird.



Una vez reiniciada, será necesario mostrar el panel sobre el que se muestra nuestra extensión. Para ello, nos dirigimos al menú “View” → “Show service pane bottom”. Es posible que por defecto muestre el panel “Album art” como se muestra en la imagen de abajo. En ese caso, debemos seleccionar el panel Muusic. Para ello, debemos hacer clic en icono  y seleccionar “Muusic”.



Tras esto, podemos probar la extensión. Se divide en dos pestañas, “Tema” y “Last.fm”:

En la **pestaña “Tema”** encontramos toda la información relacionada con el tema que reproduzcamos en Songbird:

- Datos del tema: nombre, artista, album, portada del álbum.
- Enlace para descarga del tema.
- Listado de temas similares, con enlace para descarga en los que esté disponible.
- Letras del tema.
- Fechas de conciertos.

En la **pestaña “Last.fm”** encontramos información relacionada con nuestro usuario de Last.fm. Debemos introducir un usuario en el cuadro de texto “Usuario” para que muestre información, y pinchar en el botón “Aceptar”. Opcionalmente podemos seleccionar la opción “Guardar en preferencias”, que almacenará el usuario del cuadro de texto en las preferencias para próximas ejecuciones. Una vez pinchado en el botón “Aceptar”, y tras varios segundos de espera, se mostrará:

- Un listado de los temas favoritos del usuario.
- Un listado de los temas favoritos de los amigos del usuario.
- Un listado de los temas favoritos de los vecinos del usuario.
- Un listado de los temas recientemente escuchados por el usuario registrados en el sistema Last.fm.

## 7.b. Atribuciones

---

1

**Agenda de conciertos por cortesía del portal Manerasdevivir**

Sitio web: <http://www.manerasdevivir.com>

Agenda de conciertos: <http://www.manerasdevivir.com/conciertos.php>

Distribuida bajo licencia Creative Commons Atribución 3.0 Unported

2

**Información sobre perfiles de usuario y hábitos de escucha, así como otra agenda de conciertos por cortesía de Last.fm**

Sitio web: <http://www.last.fm>

API antigua: <http://www.audioscrobbler.net/data/webservices/>

API nueva: <http://www.lastfm.es/api>

Distribuida bajo licencia Creative Commons Reconocimiento-No comercial-Compartir bajo la misma licencia 2.0 Inglaterra y País de Gales

3

**Letras de canciones por cortesía de LyricWiki**

Sitio web: <http://www.lyricwiki.org>

API: <http://lyricwiki.org/LyricWiki:Integration>

4

**Búsqueda de canciones por cortesía de Seeqpod**

Sitio web: <http://www.seeqpod.com>

API: <http://www.seeqpod.com/api.php>

5

**Comprobación de enlaces por cortesía de Link Checker del World Wide Web Consortium:**

Sitio web: <http://validator.w3.org/checklink>